

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Patent Application of:

Tomomichi OBARA, et al.

Application No.:

Group Art Unit:

Filed: April 23, 2004

Examiner:

For: AUTOMATIC TRANSACTION APPARATUS AND AUTOMATIC TRANSACTION  
SYSTEM

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN  
APPLICATION IN ACCORDANCE  
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Commissioner for Patents  
PO Box 1450  
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith  
a certified copy of the following foreign application:

Japanese Patent Application No(s). 2003-390478

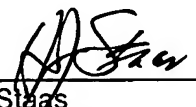
Filed: November 20, 2003

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing  
date(s) as evidenced by the certified papers attached hereto, in accordance with the  
requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: April 23, 2004

By:   
H. J. Staas  
Registration No. 22,010

1201 New York Ave, N.W., Suite 700  
Washington, D.C. 20005  
Telephone: (202) 434-1500  
Facsimile: (202) 434-1501

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日            2 0 0 3 年 1 1 月 2 0 日  
Date of Application:

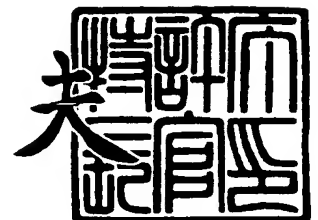
出 願 番 号            特 願 2 0 0 3 - 3 9 0 4 7 8  
Application Number:  
[ST. 10/C]:            [ J P 2 0 0 3 - 3 9 0 4 7 8 ]

出      願      人  
Applicant(s):            富士通株式会社  
                              富士通フロンテック株式会社

2 0 0 4 年   1 月 1 9 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康 夫



出証番号    出証特 2 0 0 4 - 3 0 0 0 9 0 0

【書類名】 特許願  
【整理番号】 0351685  
【提出日】 平成15年11月20日  
【あて先】 特許庁長官 殿  
【国際特許分類】 G07D 9/00  
【発明者】  
    【住所又は居所】 群馬県前橋市問屋町 1 丁目 8 番地 3 株式会社富士通ターミナル  
                                システムズ内  
    【氏名】 小原 知道  
【発明者】  
    【住所又は居所】 東京都稲城市矢野口 1 7 7 6 番地 富士通フロンテック株式会社  
                                内  
    【氏名】 富 哲  
【特許出願人】  
    【識別番号】 000005223  
    【氏名又は名称】 富士通株式会社  
【特許出願人】  
    【識別番号】 000237639  
    【氏名又は名称】 富士通フロンテック株式会社  
【代理人】  
    【識別番号】 100094514  
    【弁理士】  
    【氏名又は名称】 林 恒徳  
【選任した代理人】  
    【識別番号】 100094525  
    【弁理士】  
    【氏名又は名称】 土井 健二  
【手数料の表示】  
    【予納台帳番号】 030708  
    【納付金額】 21,000円  
【提出物件の目録】  
    【物件名】 特許請求の範囲 1  
    【物件名】 明細書 1  
    【物件名】 図面 1  
    【物件名】 要約書 1  
    【包括委任状番号】 9704944  
    【包括委任状番号】 0210821

**【書類名】 特許請求の範囲****【請求項 1】**

利用者の操作に応じて、Webサーバと通信し、案内表示と取引動作を行う自動取引装置において、

前記案内表示を行う表示ユニットと、

前記取引動作を行うための複数のIOユニットと、

前記Webサーバからの画面コンテンツに応じて、前記表示装置の画面の案内表示を制御し、且つ前記画面コンテンツに埋め込まれたオブジェクトに応じて、前記複数のIOユニットを制御する制御ユニットとを有し、

前記制御ユニットは、前記オブジェクトのスクリプトにより、前記複数のIOユニットを同期制御する前記取引動作の処理単位のメソッドを呼び出し、前記複数のIOユニットを同期制御する

ことを特徴とする自動取引装置。

**【請求項 2】**

前記制御ユニットは、前記Webサーバからの画面コンテンツを解釈し、前記案内表示を行うとともに、前記画面コンテンツに埋め込まれた前記オブジェクトのスクリプトを解釈し、前記複数のIOユニットを同期制御する前記取引動作の処理単位のメソッドを呼び出すブラウザを有し、

前記ブラウザから前記複数のIOユニットを同期制御する

ことを特徴とする請求項 1 の自動取引装置。

**【請求項 3】**

前記制御ユニットは、前記呼び出されたメソッドによるポスト依頼に応じて、前記Webサーバにリクエストを送信する

ことを特徴とする請求項 1 の自動取引装置。

**【請求項 4】**

Webサーバと、

前記Webサーバにネットワークを介し接続され、利用者の操作に応じて、前記Webサーバと通信し、案内表示と取引動作を行う自動取引装置とを有し、

前記自動取引装置は、

前記案内表示を行う表示ユニットと、

前記取引動作を行うための複数のIOユニットと、

前記Webサーバからの画面コンテンツに応じて、前記表示装置の画面の案内表示を制御し、且つ前記画面コンテンツに埋め込まれたオブジェクトに応じて、前記複数のIOユニットを制御する制御ユニットとを有し、

前記制御ユニットは、前記オブジェクトのスクリプトにより、前記複数のIOユニットを同期制御する前記取引動作の処理単位のメソッドを呼び出し、前記複数のIOユニットを同期制御する

ことを特徴とする自動取引システム。

**【請求項 5】**

前記自動取引装置の前記制御ユニットは、前記Webサーバからの画面コンテンツを解釈し、前記案内表示を行うとともに、前記画面コンテンツに埋め込まれた前記オブジェクトのスクリプトを解釈し、前記複数のIOユニットを同期制御する前記取引動作の処理単位のメソッドを呼び出すブラウザを有し、

前記ブラウザから前記複数のIOユニットを同期制御する

ことを特徴とする請求項 4 の自動取引システム。

## 【書類名】明細書

## 【発明の名称】自動取引装置及び自動取引システム

## 【技術分野】

## 【0001】

本発明は、Webサーバからの画面コンテンツに応じて、画面制御と自動取引動作とを実行する自動取引装置及び自動取引システムに関し、特に、画面情報と画面に関連する装置の制御情報を埋め込んだ画面コンテンツにより動作する自動取引装置及び自動取引システムに関する。

## 【背景技術】

## 【0002】

自動取引装置は、各種の取引に利用されており、例えば、金融分野では、自動出金機、自動入出金機が、他の分野では、自動発券機や自動発行機等が利用されている。このような自動取引装置において、近年のインターネット等のネットワークの発達に伴い、Web技術を利用して、入出金、発券、各種情報の出力を行うものが提供されている。

## 【0003】

図21は、従来の自動取引システムの構成図であり、金融業務のATM (Automatic Teller Machine) システムを示す。図21に示すように、WWW (World Wide Web) サーバ300と、現金自動取引装置400とがネットワークで接続される。

## 【0004】

サーバ300は、クライアントである自動取引装置(ATM)400のリクエストに従い、Webページ(画面コンテンツ)500をATM400に送信する。このWebページは、表示装置に表示される画面を作成するためのプログラムを、ページ記述言語(HTML, Java (登録商標) Script)を用いて作成し、この1ページ(1画面)の表示内容に関連して、駆動制御する他の装置(カード処理装置、現金処理装置、通帳処理装置、明細票処理装置等)の制御プログラムをオブジェクトとして、埋め込む。

## 【0005】

例えば、図21に示すように、Webページ500は、ページ記述言語HTML (Hyper Text Markup Language)により、画面作成プログラムと、オブジェクト(アプレット)を指定するアプレットタグと、その実行方法(method)をScriptで指定する画面コンテンツ502と、オブジェクト(アプレット)のmethodを実行するプログラムをセットしたアプレット510からなる。

## 【0006】

このWebページ500は、WWWサーバ300からATM400のブラウザ410にダウンロードされる。一方、ATM400は、カーネル(OS)の制御の基に、ATMミドルウェアが動作し、I/O動作(取引動作)する。ATM400は、I/Oメカユニットとして、カードリーダー/ライターユニット440、レシート/ジャーナルプリンタ441、紙幣/硬貨処理ユニット442、通帳処理ユニット443、顧客操作パネル等を有する。

## 【0007】

ブラウザ410は、Webページの画面作成プログラムに従い、画面を顧客操作パネルに表示するとともに、画面コンテンツ502のアプレットタグと、method名を解析し、アプレット510の該当プログラムを実行し、I/Oユニット440～443にコマンドを発行する。

## 【0008】

このようなWebブラウザによるATM制御において、図21及び図22に示すように、methodを、各装置(現金処理装置等)単位とし、画面コンテンツ502の埋め込みオブジェクト(アプレット)として、装置インタフェース振分け部(Machine ID)を指定し、画面コンテンツ502のScript (Java script)で、各装置の動作(この例では、Initialize)方法を指定することが提案されている(例えば、特許文献1参照)。

## 【0009】

この方法では、画面コンテンツ502のアプレットタグで、装置インタフェース振分け部420が指定され、そして、装置振分け部420は、Scriptを順次読み込み、解釈し、動作指令を対応するインタフェース部430～433に振分け、対応するI/Oユニット440～443を動作する。

【0010】

この例では、装置初期化の動作指令であるから、図23に示すように、画面コンテンツ502のScriptのInitialize("Cash")を読み込み、現金処理装置442のコントローラに初期化コマンドを送り、初期化完了応答を得て、次に、ScriptのInitialize("Card")を読み込み、カード処理装置440のコントローラに初期化コマンドを送り、初期化完了応答を得る。以下、順次、ScriptのInitialize( )を読み込み、対応する処理装置のコントローラに初期化コマンドを送り、初期化完了応答を得て、終了する。

【0011】

この提案では、同一機能のATMをWeb制御する場合に、画面コンテンツの1つのアプレットタグで、複数のユニットを動作することができ、画面コンテンツのHTML内の記述を少なくでき、埋め込みオブジェクトを軽くできる。

【特許文献1】特開2000-298752号公報（図6、図7）

【発明の開示】

【発明が解決しようとする課題】

【0012】

しかしながら、ATMは、同一の機能のものならず、異なる機能のATMも多く存在する。例えば、現金処理機能として、紙幣及び硬貨を処理するものと、紙幣のみを処理するものがあり、又、通帳処理を行うものと、行わないものがあり、更に、入出金処理するものと、出金専用のものがある。

【0013】

このような異なる機能のATMをWeb制御する場合には、この従来技術では、呼び出されるmethodが、各ユニット単位であるため、構成機能又はユニットに応じて、同一のアプレットタグでも、画面コンテンツのScript自体を大幅に変更する必要がある。即ち、画面コンテンツの記述を、制御するATMの構成機能又はユニットの相違に応じて、設計する必要がある。

【0014】

このため、WWWサーバ300での、Webページの開発に多大の労力を要し、且つ新しい機能（例えば、携帯電話との関係機能等）を付加するのに、手間がかかるおそれがある。

【0015】

又、従来技術では、呼び出されるmethodが、各ユニット単位であるため、画面コンテンツのScriptをユニット単位に設ける必要がある。このため、前述の図23で示した初期化の例のように、複数ユニットの同期制御を行うには、Script単位にコマンド発行及び応答受信を行うため、同期制御に時間がかかり、自動取引装置の利用者の利用時間が長くなり、稼働率の低下及び応答待ち時間の長期化を招くおそれがある。

【0016】

従って、本発明の目的は、画面コンテンツ内のオブジェクトの記述の変更を少なくして、複数の異なる機能の自動取引装置を制御するための自動取引装置及び自動取引システムを提供することにある。

【0017】

又、本発明の他の目的は、複数の異なる機能の自動取引装置を制御する場合に、WebサーバのWebページの作成を簡単化するための自動取引装置及び自動取引システムを提供することにある。

【0018】

更に、本発明の他の目的は、Web制御しても、複数の異なるユニットの同期制御の速度を向上し、応答待ち時間を短縮するための自動取引装置及び自動取引システムを提供す

ることにある。

【課題を解決するための手段】

【0019】

この目的の達成のため、本発明の自動取引装置は、利用者の操作に応じて、Webサーバと交信し、案内表示と取引動作を行う自動取引装置において、前記案内表示を行う表示ユニットと、前記取引動作を行うための複数のIOユニットと、前記Webサーバからの画面コンテンツに応じて、前記表示装置の画面の案内表示を制御し、且つ前記画面コンテンツに埋め込まれたオブジェクトに応じて、前記複数のIOユニットを制御する制御ユニットとを有し、前記制御ユニットは、前記オブジェクトのスクリプトにより、前記複数のIOユニットを同期制御する前記取引動作の処理単位のメソッドを呼び出し、前記複数のIOユニットを同期制御する。

【0020】

又、本発明の自動取引システムは、Webサーバと、前記Webサーバにネットワークを介し接続され、利用者の操作に応じて、前記Webサーバと交信し、案内表示と取引動作を行う自動取引装置とを有し、前記自動取引装置は、前記案内表示を行う表示ユニットと、前記取引動作を行うための複数のIOユニットと、前記Webサーバからの画面コンテンツに応じて、前記表示装置の画面の案内表示を制御し、且つ前記画面コンテンツに埋め込まれたオブジェクトに応じて、前記複数のIOユニットを制御する制御ユニットとを有し、前記制御ユニットは、前記オブジェクトのスクリプトにより、前記複数のIOユニットを同期制御する前記取引動作の処理単位のメソッドを呼び出し、前記複数のIOユニットを同期制御する。

【0021】

更に、本発明では、好ましくは、前記制御ユニットは、前記Webサーバからの画面コンテンツを解釈し、前記案内表示を行うとともに、前記画面コンテンツに埋め込まれた前記オブジェクトのスクリプトを解釈し、前記複数のIOユニットを同期制御する前記取引動作の処理単位のメソッドを呼び出すブラウザを有し、前記ブラウザから前記複数のIOユニットを同期制御する。

【0022】

更に、本発明では、好ましくは、前記制御ユニットは、前記呼び出されたメソッドによるポスト依頼に応じて、前記Webサーバにリクエストを送信する。

【0023】

更に、本発明では、好ましくは、前記制御ユニットは、ページ記述言語で記述された画面作成プログラムと前記オブジェクトのスクリプトと、前記スクリプトで呼び出されるメソッドプログラムとを有する画面コンテンツを、前記Webサーバから受信する。

【0024】

更に、本発明では、好ましくは、前記制御ユニットは、前記呼び出したメソッドにより、前記複数のIOユニットに動作コマンドを発行し、前記IOユニットからの応答を受信する。

【0025】

更に、本発明では、好ましくは、前記複数のIOユニットは、少なくとも、現金処理ユニットと、カード処理ユニットとを有する。

【0026】

更に、本発明では、好ましくは、前記制御ユニットは、前記スクリプトに付記された入力パラメータに従い、前記メソッドにより同期制御する複数のIOユニットを指定する。

【0027】

更に、本発明では、好ましくは、前記ブラウザは、前記画面コンテンツのページ記述言語で記述された画面作成プログラムにより前記案内画面を作成するとともに、前記オブジェクトのスクリプトから、前記メソッドプログラムを呼び出し、前記複数のIOユニットを同期制御する。

【0028】

更に、本発明では、好ましくは、前記ブラウザは、前記画面コンテンツのページ記述言語で記述された画面作成プログラムにより前記案内画面を作成するとともに、前記オブジェクトのスクリプトのアプレット指定とメソッド指定から、前記アプレットのメソッドプログラムを呼び出し、前記複数の I O ユニットの同期制御する。

【0029】

更に、本発明では、好ましくは、前記ブラウザは、前記呼び出したメソッドにより、前記 I O ユニットの各々を制御する複数の I O コントローラに、動作コマンドを発行し、前記 I O コントローラからの応答を受信する。

【発明の効果】

【0030】

本発明では、Web サーバの画面コンテンツに取引処理の各処理単位のオブジェクトを埋め込み、処理単位の 1 つのメソッドを呼び出すことにより、複数の I O ユニットの動作制御でき、一般的な取引処理の流れで、異なる自動取引装置に共通に使用できる。このため、異なる機能、構成の自動取引装置を Web 制御する画面コンテンツの作成が容易となる。

【0031】

又、1 つのメソッドをコールするため、並列制御が可能となり、高速の I O ユニットの制御を実現できる。これにより、利用者の自動取引装置の待ち時間を短縮でき、且つ稼働率を向上できる。

【発明を実施するための最良の形態】

【0032】

以下、本発明の実施の形態を、自動取引システム、Web 制御による I O 制御機構、自動取引処理、既存ミドルウェアのカスタマイズ化方法、他の実施の形態の順で説明する。

【0033】

〔自動取引システム〕

図 1 は、本発明の一実施の形態の自動取引装置の外観図、図 2 は、図 1 の自動取引装置のブロック図、図 3 は、本発明の一実施の形態の自動取引システムのシステム構成図である。

【0034】

図 1 に示すように、自動取引装置 1 は、磁気カードの挿入及び排出のためのカード出入口 2、磁気通帳の挿入及び排出のための通帳出入口 3、紙幣の投入及び排出のための紙幣出入口 4、硬貨の投入及び排出のための硬貨出入口 5、利用者の操作のための U O P (User Operation Panel) 6、利用者に運用状況を表示するための運用表示器 7、利用者を検出する顧客センサ 8 を備える。

【0035】

図 2 は、図 1 の自動取引装置 1 のブロック図である。C R W (Card Reader Writer) ユニット 10 は、カード出入口 (カード挿入口) 2 から挿入された磁気カードを、図示しない搬送機構で搬送しながら、磁気ヘッドで読み取り、出入口 2 へ返却する。C R W ユニット 10 には、イメージセンサが設けられ、磁気カード (エンボス部) を光学的に読み取る。

【0036】

R P R (Receipt Printer) ユニット 20 は、印字ヘッドで、レシート用紙に、取引結果の印字を行い、レシートをカード出入口 2 に排出する。又、R P R ユニット 20 は、排出されたレシートが所定時間内に利用者に抜き取られない時に、出入口 2 から戻されたレシートを収容する。

【0037】

J P R (Journal Printer) ユニット 70 は、印字ヘッドで、ジャーナル用紙に、取引状態、結果等を印字する。U O P (User Operation Panel) ユニット 30 は、U O P (タッチパネル付きディスプレイ) 6 とその制御回路で構成される。通帳 (P P R) 処理ユニット 40 は、通帳出入口 3 から挿入された磁気通帳を読み取り、且つ磁気通帳に取引印字



を行い、通帳出入口3から排出する。

【0038】

紙幣／硬貨処理ユニット50は、紙幣出入口4、硬貨出入口5から投入された紙幣、硬貨を鑑別、計数し、スタッカに収容する入金動作を行い、且つ現金スタッカから要求された紙幣、硬貨を取り出し、紙幣出入口4、硬貨出入口5へ放出する出金動作を行う。

【0039】

制御部60は、LAN等のネットワーク90を介し、これら制御ユニット10、20、30、40、50、70に接続され、図3にて後述するソフトウェアの構成により自動取引処理を行う。

【0040】

図3は、本発明の一実施の形態の自動取引システムのブロック図である。自動取引装置1は、インターネット等のネットワーク110を介し、WWW (World Wide Web)サーバ(ホスト)100と、取引処理に必要なコマンド、パラメータ、データのやりとりを行う。

【0041】

自動取引装置1では、前述の制御部60が、ブラウザ120と、ATMミドルウェア130と、カーネル(OS)140と、デバイスドライバ150とを搭載する。

【0042】

デバイスドライバ150は、カード(CRW)ユニット10をドライブするためのカードユニットドライバ151と、レシート／ジャーナルユニット(RPR/JPR)20、70をドライブするためのレシート／ジャーナルユニットドライバ152と、BRU(紙幣)ユニット50をドライブするためのBRUドライバ153と、CRU(硬貨)ユニット50をドライブするためのCRUドライバ154と、UOP30をドライブするためのグラフィックドライバ155及びタッチスクリーンドライバ156と、通帳ユニット40をドライブするためのPPRドライバ157とを有する。

【0043】

ブラウザ120は、例えば、Internet Explorer (Microsoft社商標名)等のWebブラウザで構成され、Webサーバ100にコンテンツの送信を要求し、Webサーバ100が送信する画面コンテンツ(Webページ)を解釈し、表示等を行う。ここでは、HTMLとJava Scriptで構成された取引処理に必要なWebページを要求し、送信されたWebページを解釈し、ATMミドルウェア130、UOP30の画面制御を行う。

【0044】

カーネル140は、例えば、Windows (登録商標)、Linux等の周知のOS (Operating System)で構成され、カーネル140の動作環境下で、ブラウザ120、ATMミドルウェア130、デバイスドライバ150が動作する。

【0045】

ATMミドルウェア130は、パラメータファイル160と、IOコントロール層170と、IOクライアント層190と、IOサーバ層200と、IOサービスプロバイダ層210とを有する。

【0046】

このIOクライアント層190は、装置に実装された個々のI/Oユニットを制御するものであり、IOサーバ層200は、IO動作の起動、終結及び通信プロトコル制御を行うものであり、IOサービスプロバイダ層210は、各I/Oユニットへの電文の変換を行うものである。これらは、その装置の機能範囲、種類、接続されたI/Oユニットの仕様に従い、設計された既存のミドルウェア180である。

【0047】

一方、IOコントロール層170は、Webサーバ100とミドルウェア共通アプリケーションインタフェース規約で、コマンド、データの送受信を行う。この共通アプリケーションインタフェース(API)は、装置の機種によって、機能範囲が異なるため、全ての機種を動作できる共通のコマンド、データ体系で構成され、図6で後述する。

## 【0048】

IOコントロール層170は、IOクライアント層190のアプリケーションインタフェース(API)を統合、集約し、抽象度の高い共通APIを構築するものである。パラメータファイル160は、ベンダ(ATMメーカ)固有のシステム仕様によって、一意に決定される入力パラメータ/固定パラメータを格納するものである。

## 【0049】

IOコントロール層170は、IOクライアント層190の呼び出し時に、このパラメータファイル160から各IOクライアント層固有のパラメータを呼び出し、共通APIを、既存のクライアントAPIに変換する。

## 【0050】

これにより、抽象度の高い共通APIを、自動取引装置1のATMミドルウェア190、搭載IOユニットの種類に合致したクライアントAPIに変換でき、既存のATMミドルウェア180とIOユニットを動作することができる。即ち、既存のATMミドルウェアを、共通APIで動作するように、カスタマイズ化できる。

## 【0051】

図3に示すように、本発明では、Webページの画面コンテンツ122のアプレットとして、ATMの処理単位で定義したエージェント(Agent)123を埋め込み、処理単位で処理に必要なIOユニットの動作を制御する。詳細は、図4以下で後述するが、アプレット名を、同期制御や、同期制御の初期化制御という処理単位で定義し(以下、Agentという)、methodも処理単位で設ける。即ち、処理に対応したIOユニットの動作を制御するmethodを有するアプレット(Agent)を設け、処理単位で、アプレット(Agent)、methodを指定する。

## 【0052】

これにより、複数ユニットを同期動作する場合でも、処理単位にくることができるため、且つ処理単位のアプレット名を付すことができるため、異なる機能のATMのWeb制御をWebページの変更箇所が少なくでき、且つ複雑な自動取引制御においても、Webページを容易に作成できる。

## 【0053】

又、1つのmethodをコールするだけで、複数のIOユニットを制御できるため、複数のIOユニットを並行制御でき、それだけ、処理速度を向上でき、Web制御しても、ATMの制御時間を短縮でき、利用者の待ち時間を短縮できる。

## 【0054】

[Web制御によるIO制御機構]

次に、Web制御によるIO制御機構を説明する。図4は、図3の構成におけるWebブラウザとIOコントローラのブロック図、図5は、図3の構成におけるATMミドルウェアの構成図、図6は、図3及び図4における共通APIのコマンドの説明図である。

## 【0055】

図4に示すように、Webブラウザ120には、画面コンテンツ(HTML+JavaScript)122と、Agent(アプレット)123とが、WWWサーバ100からダウンロードされる。Agent123は、各ユニットを制御するための制御Agent群124と、ポスト処理を行うPOSTAgent126と、画面構築のためのViewAgent128と、Java Native Interfaceと共通APIインタフェース132を介しIOコントローラ170と通信するための通信DLL群(Dynamic Link Library)129を有する。

## 【0056】

尚、インタフェースが同一であれば、通信DLL群129は、不要である。制御Agent群124は、表示画面で必要な制御Agentのみダウンロードされる。図7以下で詳細に説明するように、制御Agentは、ATM取引の処理単位に構成されている。

## 【0057】

図4の構成の動作を説明すると、画面コンテンツ122のJava Scriptで記述されたメ

ソッド名で、制御 Agent 124 のメソッドを呼び出す。呼び出されたメソッドは、通信 DLL 群 129 を介し IO コントローラ 170 にコマンドを発行する。

【0058】

IO コントローラ 170 は、ATM ミドルウェア 180 を介し対応するユニットを動作し、動作完了を受ける。IO コントローラ 170 は、コマンドの実行結果を制御 Agent 124 に応答する。制御 Agent 124 は、Post 処理を行う POST Agent 126 又は画面更新を行う View Agent 128 に POST 依頼又は描画依頼を依頼し、POST 処理又は画面更新処理を行わせる。

【0059】

この Agent を説明する前に、先ず、共通 API を説明する。図 6 は、共通 API のコマンド種類の一例を示す。CRW (カードリーダー/ライター) コマンドとしては、カード挿入コマンド、カード排出コマンドが用意されている。

【0060】

RPR (レシートプリンタ) コマンドとしては、印字コマンド、放出コマンド等が用意されている。PPR (通帳プリンタ) コマンドとしては、通帳挿入コマンド、印字コマンド、MS (磁気ストライプ) 書き込みコマンド、通帳排出コマンド、オートターンページコマンド等が用意されている。

【0061】

BRU (紙幣リサイクルユニット) コマンドとしては、初期化コマンド、受付/計数コマンド、収納コマンド、入金返却コマンド、繰り出しコマンド、放出コマンド、取り込みコマンド、搬送路チェックコマンド、ジャムリセットコマンド等が用意されている。CRU (硬貨リサイクルユニット) コマンドも同様であり、説明を省略する。

【0062】

次に、図 3 及び図 4 の ATM ミドルウェア 130 の構成を、図 5 により説明する。IO コントロール層 170 は、各 IO のコントロールを行う IO コントロールライブラリ群 (IO コントローラ) 171~178 を有する。

【0063】

ここでは、IO コントロールライブラリ群は、通帳コントロールライブラリ 171、CRW コントロールライブラリ 172、テンキーコントロールライブラリ 173、レシートコントロールライブラリ 174、紙幣コントロールライブラリ 175、硬貨コントロールライブラリ 176、ジャーナルコントロールライブラリ 177、取引制御コントロールライブラリ 178 を有する。

【0064】

これらコントロールライブラリ 171~178 は、共通 API で指定されたタスク (カードコントロール EXE 等) により呼び出され、そのタスクを前述のパラメータテーブル 160 を利用して、既存ミドルウェアのクライアント API に変換する。

【0065】

又、既存ミドルウェア 180 の IO クライアント層 190 は、装置に実装された個々の I/O ユニットの制御するものであり、ここでは、カード (CRW) クライアント 191、硬貨クライアント 192、紙幣クライアント 193、RPR クライアント 194、JPR クライアント 195、PPR クライアント 196 等が設けられている。

【0066】

同様に、IO サーバ層 200 も、個々の IO 動作の起動、終結及び通信プロトコル制御を行うため、個々の IO に分割されている。即ち、カード (CRW) サーバ 201、硬貨サーバ 203、紙幣サーバ 202、RPR (レシートプリンタ) サーバ 204、JPR サーバ 205、PPR (通帳プリンタ) サーバ 206 が設けられている。

【0067】

又、同様に、IO サービスプロバイダ層 210 も、各 IO ユニットへの電文の変換を行うため、個々の IO に分割されている。即ち、カード (CRW) サービスプロバイダ 211、硬貨サービスプロバイダ 213、紙幣サービスプロバイダ 212、RPR サービスプ

ロバイダ 214、JPR サービスプロバイダ 215、PPR サービスプロバイダ 216 が設けられている。

#### 【0068】

即ち、ATM ミドルウェアを構成する、コントロールライブラリ、クライアント、サーバ、サービスプロバイダが、個々の I/O ユニットに対応して設けられ、要求された共通 API のコマンド、パラメータを、I/O コントロール 170 が、既存ミドルウェア API のコマンド、パラメータに変換し、既存ミドルウェアを介し I/O ユニットの動作する。

#### 【0069】

次に、前述の Agent を、図 7、図 8 で説明する。図 7 に示すように、同期 Agent は、複数の I/O ユニットの同期制御するプログラムであり、メソッド（プログラム）として、初期化、メカリセット、紙幣／硬貨挿入、媒体同時排出、印字／繰り出し／MS ライト／排出準備、入金返却、収納、強制排出／取込み、ユニット情報取得／取引状態設定／二面表示制御、入出金準備、強制補充、ジャムリセット、カード／通帳挿入等を有する。

#### 【0070】

各メソッドは、図 7 の黒丸で示す I/O コントローラ（即ち、I/O ユニット）にコマンドを発行し、コマンド実行結果の応答を受ける。例えば、初期化メソッドは、紙幣コントローラ 175、硬貨コントローラ 176、通帳コントローラ 171、カードコントローラ 172、レシートコントローラ 174、ジャーナルコントローラ 177、取引制御コントローラ 178、テンキーコントローラ 173（図 5 参照）に、初期化コマンドを発行し、各コントローラによる初期化処理を行わせ、各コントローラから初期化処理結果を応答として、受信する。

#### 【0071】

同様に、メカリセットメソッドは、紙幣コントローラ 175、硬貨コントローラ 176、通帳コントローラ 171、カードコントローラ 172、レシートコントローラ 174、ジャーナルコントローラ 177（図 5 参照）に、メカリセットコマンドを発行し、各コントローラによるメカリセット処理を行わせ、各コントローラからメカリセット処理結果を応答として、受信する。

#### 【0072】

以下、同様に、紙幣／硬貨挿入メソッドは、紙幣コントローラ 175、硬貨コントローラ 176（図 5 参照）に、挿入コマンドを発行し、各コントローラによる挿入処理を行わせ、各コントローラから挿入処理結果を応答として、受信する。

#### 【0073】

同様に、POST エージェントとして、POST 処理メソッドと、POST データ保持メソッドを設ける。又、View（テキスト表示）エージェントとして、フォント設定メソッド、テキスト表示メソッド、テキスト消去メソッドを設ける。

#### 【0074】

更に、図 8 に示すように、単一の I/O ユニットの制御するアプレットも、同一アーキテクチャで制御するため、Agent として、定義する。即ち、図 8 に示すように、紙幣制御 Agent、通帳制御 Agent、カード制御 Agent、レシート制御 Agent、取引制御 Agent、テンキー Agent を設ける。

#### 【0075】

そして、紙幣制御 Agent は、各々紙幣コントローラ 175 を制御する受付／計数メソッド、収納メソッド、入金返却メソッド、放出メソッド、キャンセルメソッドを有する。又、通帳制御 Agent は、各々通帳コントローラ 171 を制御する行セット／ページマークリードメソッド、MS（磁気ストライプ）リードメソッド、オートターンページメソッド、ページチェックオートターンメソッド、通帳構成情報設定メソッドを有する。

#### 【0076】

更に、カード制御 Agent は、各々カード（CRW）コントローラ 172 を制御するカード挿入メソッド、キャンセルメソッド、振込みカード印字メソッド、振込みカード発

行メソッド、排出準備メソッドを有する。レシート制御Agentは、各々レシートコントローラ174を制御するオーバーレイ登録メソッド、排出準備メソッドを有する。

#### 【0077】

同様に、取引制御Agentは、各々取引制御コントローラ178を制御する取引情報設定メソッド、装置状態監視メソッド、装置状態取得メソッド、運用情報設定メソッド、キャンセルメソッドを有する。テンキーAgentは、各々テンキーコントローラ173を制御するテンキー入力開始メソッド、テンキー入力終了メソッドを有する。

#### 【0078】

このようなAgent及びメソッドを呼び出すWeb操作を、図9乃至図12で説明する。図9は、Webサーバ100が、送信する画面コンテンツの説明図であり、図10は、図9の画面コンテンツによるIO制御の説明図である。

#### 【0079】

図9に示すように、HTML（ページ記述言語）で記述された画面コンテンツ内で、画面内で呼び出すAgent名を指定する。ここでは、一例として、アプレットタグ<APPLET CODE="U\_\_agtSync.class">により、同期制御（Sync）エージェントを呼び出すことを示す。

#### 【0080】

更に、<SCRIPT Language="JavaScript">の記述により、JavaScriptで、同期制御エージェントのメソッドの呼び出しを指定する。即ち、ret=document.U\_\_agtSync.initの記述で、同期制御エージェントの初期化メソッドの呼び出しを指定する。尚、ここでは、画面表示コンテンツの詳細を省略してある。

#### 【0081】

この画面コンテンツが、ブラウザ120にダウンロードされると、図10に示すように、アプレットタグにより、同期制御エージェントが指定され、且つSCRIPTの呼び出しメソッド名により、初期化メソッドが呼び出される。

#### 【0082】

前述のように、呼び出された初期化メソッドは、紙幣コントローラ175、硬貨コントローラ176、通帳コントローラ171、カードコントローラ172、レシートコントローラ174、ジャーナルコントローラ177、取引制御コントローラ178、テンキーコントローラ173（図5参照）に、初期化コマンドを発行し、各コントローラによる初期化処理を行わせ、各コントローラから初期化処理結果を応答として、受信する。

#### 【0083】

この場合に、図12に示すように、呼び出された初期化メソッドは、紙幣コントローラ175、硬貨コントローラ176、通帳コントローラ171、カードコントローラ172、レシートコントローラ174、ジャーナルコントローラ177、取引制御コントローラ178、テンキーコントローラ173（図5参照）に、初期化コマンドを連続的に発行し、これらコントローラからの応答を、処理完了したものから順次受ける。

#### 【0084】

このため、IOユニットを並列的に制御でき、複数のIOユニットを制御しても、短時間で制御できる。これに比し、図23で示したように、従来のScriptで、IOユニット単位のメソッドを指定する方法では、Script毎に、コマンド発行、応答受信を順次行う順次制御のため、WebによるIO制御に時間がかかる。

#### 【0085】

更に、前述のAgentは、処理単位名が付された処理単位で設けられるため、この処理単位の内容を若干変更するだけで、各種の機能のATMに、同一のアプレットタグ、メソッド名で、対応できる。この例では、前述のメソッド呼び出しSCRIPTのメソッド名の括弧内に設けられたinput Paramの変更により実現している。即ち、図11に示すように、input Paramは、各IOコントローラ（紙幣、硬貨、通帳、カード、レシート、ジャーナル、取引制御、テンキー）に、入力フラグを設定する数バイトのビット情報（図では、8ビットの情報）で構成されている。

## 【0086】

入力フラグは、"1" がセットされた I O コントローラへのメソッドの入力を許可することを示す。呼び出されたメソッドは、この入力フラグを参照して、コマンドを発行する I O コントローラを決定する。このため、入力フラグを Web サーバ 100 で操作することにより、異なる機能の ATM でも、同一のアプレットタグ、メソッド名で、同一の処理を実行させることができる。

## 【0087】

例えば、硬貨を扱わない取引装置では、硬貨の入力フラグを"0" にセットしておけば、硬貨コントローラへのコマンド発行を防止できる。同様に、通帳を扱わない取引装置では、通帳の入力フラグを"0" にセットしておけば、通帳コントローラへのコマンド発行を防止できる。

## 【0088】

又、SCRIPT のメソッド名の括弧内に設けられた ParseInt (postMode) により、POST エージェントを、このエージェント内で指定している。このため、POST 処理を、エージェント主導で円滑に実行できる。又、エージェントが全てのコマンド完了を受信したタイミングで、直ちに、POST することができる。

## 【0089】

## [自動取引処理]

次に、図 13 のカード入金取引を例に、Web サーバ 100 の画面コンテンツと、顧客操作画面 (UOP 画面) と、Agent とメソッド、I O コントローラとの関係を説明する。

## 【0090】

Web サーバ 100 は、JSP (Java Server Pages) により取引種別選択の画面コンテンツ及びアプレットを、ATM1 に発行する。ATM1 では、ブラウザ 120 で、取引種別選択画面を、UOP 6 の顧客操作画面に、表示する。

## 【0091】

そして、画面コンテンツに埋め込まれたアプレット名とメソッド名により、図 7 の同期制御 Agent の入出金準備メソッドを呼び出し、紙幣コントローラ 175 と、硬貨コントローラ 176 に、入出金準備コマンドを発行し、入出金の準備を行わせる。両コントローラ 175、176 は、入出金の準備が完了すると、入出金準備メソッドに完了を応答する。

## 【0092】

UOP 6 の種別選択画面で、入金キーが押下されると、Web サーバ 100 に報告される。これにより、Web サーバ 100 は、JSP により、カード開始処理に移行し、挿入処理の画面コンテンツを ATM1 に発行する。ATM1 では、ブラウザ 120 で、UOP 6 の顧客操作画面に、カード挿入画面を表示する。

## 【0093】

そして、画面コンテンツに埋め込まれたアプレット名カード制御、メソッド名カード挿入により、カード制御 Agent のカード挿入メソッドを呼び出し、カードコントローラ 172 にカード挿入コマンドを発行する。コントローラ 172 は、カード挿入コマンドにより、カードユニット 10 を動作する。カードユニット 10 がカードの挿入を検出し、カードを読取ると、カードコントロールライブラリ 172 より、カード挿入完了を、カード挿入メソッドに応答する。

## 【0094】

そして、カード挿入メソッドは、取引状態 (ここでは、カード挿入検出及びカードの読取りデータ) を POST 依頼し、POST エージェント 126 が、Web サーバ 100 にリクエスト送信する。この場合、同期制御 Agent のカード/通帳挿入メソッドを呼び出し、その入力フラグ (図 11 参照) で、カードコントローラに入力フラグを設定しておくことにより、同様に、制御できる。

## 【0095】

次に、Webサーバ100のJSPは、カード挿入処理の終了によるリクエストにより、現金挿入処理の画面コンテンツをATM1に発行する。ATM1では、ブラウザ120が、画面コンテンツに応じて、UOP6の顧客操作画面に、現金挿入画面を表示する。そして、画面コンテンツに埋め込まれたアプレット名であるカード制御Agent、メソッド名であるカード挿入により、同期制御Agentの紙幣/硬貨挿入メソッドを呼び出し、紙幣コントローラ175、硬貨コントローラ176に受付計数コマンドを発行する。

#### 【0096】

紙幣コントローラ175、硬貨コントローラ176は、受付計数コマンドにより、紙幣ユニット、硬貨ユニット50を動作する。紙幣、硬貨ユニット50は、挿入口を開き、投入紙幣、硬貨を検出し、挿入口を閉じ、投入紙幣、硬貨を計数する。計数を終了すると、紙幣コントローラ175と硬貨コントローラ176より、受付計数完了を、紙幣/硬貨挿入メソッドに応答する。そして、紙幣/硬貨挿入メソッドは、取引状態（ここでは、受付枚数、合計額）をPOST依頼し、POSTエージェント126が、Webサーバ100にリクエスト送信する。

#### 【0097】

次に、Webサーバ100のJSPは、現金挿入処理の終了によるリクエストにより、残高更新等の入金処理に移行し、コンピュータ処理の画面コンテンツをATM1に発行し、UOP6の顧客操作画面に、お待ち下さい画面を表示する。

#### 【0098】

そして、ブラウザ120は、画面コンテンツに埋め込まれたアプレット名である同期制御Agent、メソッド名である印字/繰り出し/MSライト/排出準備により、同期制御Agentの印字/繰り出し/MSライト/排出準備メソッド(図7参照)を呼び出し、カードコントローラ172にMSライトコマンド、排出準備コマンドを、レシートコントローラ174に印字コマンド、排出準備コマンドを、ジャーナルコントローラ177に印字コマンドを発行する。

#### 【0099】

これにより、カードユニット10は、カードの磁気ストライプに書き込みを行い、カードの排出を準備し、レシートプリンタ20は、レシートを印字し、排出の準備を行い、ジャーナルプリンタ70は、ジャーナルに印字を行う。各コントローラ172, 174, 177は、コマンド実行完了により、印字/繰り出し/MSライト/排出準備メソッドに完了を応答する。

#### 【0100】

そして、印字/繰り出し/MSライト/排出準備メソッドは、取引状態（ここでは、排出準備完了）をPOST依頼し、POSTエージェント126が、Webサーバ100にリクエスト送信する。

#### 【0101】

Webサーバ100のJSPは、媒体排出処理に移行し、媒体排出の画面コンテンツをATM1に発行する。ATM1では、ブラウザ120が、UOP6の顧客操作画面に、媒体排出画面を表示する。そして、ブラウザ120は、画面コンテンツに埋め込まれたアプレット名である同期制御Agent、メソッド名である媒体排出により、同期制御Agentの媒体排出メソッド(図7参照)を呼び出し、カードコントローラ172に放出コマンドを、レシートコントローラ174に放出コマンドを、図示しないサウンドコントローラに、ベル再生コマンドを発行する。

#### 【0102】

これにより、カードユニット10は、カードを、レシートプリンタ20は、レシートを放出し、カード、レシートの抜き取りを検出し、所定時間内に抜き取られないと、連続音を発する。各コントローラ172, 174は、コマンド実行完了により、媒体排出メソッドに完了を応答する。

#### 【0103】

そして、媒体排出メソッドは、取引状態（ここでは、抜き取り完了）をPOST依頼し

、POSTエージェント126が、Webサーバ100にリクエスト送信する。

【0104】

次に、Webサーバ100のJSPは、ATM1からの抜き取り完了によるリクエストを受け、取引終了処理に移行し、取引終了の画面コンテンツをATM1に発行する。ATM1では、ブラウザ120により、UOP6の顧客操作画面に、終了画面を表示する。ブラウザ120は、画面コンテンツに埋め込まれたアプレット名である同期制御Agent、メソッド名である紙幣／硬貨収納により、同期制御Agentの紙幣／硬貨メソッド(図7参照)を呼び出し、紙幣コントローラ175と硬貨コントローラ176に収納コマンドを発行する。

【0105】

これにより、紙幣ユニット、硬貨ユニット50は、計数紙幣、硬貨を内部スタッカに収納する。各コントローラ175、176は、コマンド実行完了により、紙幣／硬貨メソッドに完了を応答する。そして、紙幣／硬貨メソッドは、取引状態(ここでは、収納完了)をPOST依頼し、POSTエージェント126が、Webサーバ100にリクエスト送信する。

【0106】

これにより、Webサーバ100は、前述の取引種別選択処理に戻り、同様な処理を繰り返す。

【0107】

このように、Webサーバ100の画面コンテンツに取引処理の各処理単位のAgentを埋め込み、処理単位の1つのメソッドを呼び出すことにより、複数のIOユニットを動作制御でき、一般的な取引処理の流れで、異なる自動取引装置に共通に使用できる。このため、異なる機能、構成の自動取引装置をWeb制御する画面コンテンツの作成が容易となり、且つ並列制御が可能となり、高速のIOユニット制御を実現できる。これにより、利用者の自動取引装置の待ち時間を短縮でき、且つ稼働率を向上できる。

【0108】

この例では、カード入金処理で説明したが、出金処理、通帳を使用した入金処理、出金処理、記帳処理、残高照会等も同様である。

【0109】

[既存ATMミドルウェアのカスタマイズ化方法]

次に、ATMの機種により、IOユニットの仕様や、ATMミドルウェアの仕様が異なる場合でも、この共通インタフェースで動作できる方法を説明する。この場合に、ATMミドルウェア自体を、この共通インタフェースに合わせて設計することは、共通インタフェースを制定した意味が薄れる。

【0110】

このため、IOユニットを動作するための既存のATMミドルウェアを活かして、共通インタフェースで動作できるように、既存ATMミドルウェアのカスタマイズを行う。

【0111】

図14及び図15は、本発明の一実施の形態の動作説明図であり、図16は、その共通インタフェースとベンダ固有インタフェースの説明図、図17及び図18は、IOコントロールライブラリでのインタフェース変換処理フロー図である。

【0112】

図14及び図15では、サーバ100のATMアプリケーションをWebアプリケーションで示し、ブラウザ120で、JAVA(登録商標)のATMアプレット関数呼び出しの形式でCRW(カードユニット)コマンド、BRU(紙幣ユニット)コマンドを、パラメータa、b、A、Bを付して、共通インタフェースとして、発行する。

【0113】

一方、ATM1では、図5で説明したように、パラメータファイル160は、各ベンダのシステム仕様によって、一意に決定される入力パラメータ／固定パラメータを格納している。図14では、A社固有の設定情報として、CRWコマンドAに対し、固有パラメー



タ c, d を、BRU コマンド X に対し、固有パラメータ C, D, E を格納する。

【0114】

同様に、図 15 では、A 社固有の設定情報として、CRW コマンド A に対し、固有パラメータ c を、BRU コマンド X に対し、固有パラメータ C, D, E, F を格納する。

【0115】

IO コントロール層 170 は、IO クライアント層 190 を含む ATM ミドルウェア 180 の呼び出し時に、このパラメータファイル 160 から各 IO クライアント層固有のパラメータを呼び出し、共通 API を、既存のクライアント API に変換する。

【0116】

例えば、図 14 では、IO コントロール層 170 が、共通インタフェース (API) の CRW コマンド A (a, b) を、CRW ミドル API のコマンド A (a, b, c, d) に変換し、CRW ミドルウェア 180 に送り、CRW ユニット 10 を動作し、共通インタフェース (API) の BRU コマンド X (A, B) を、BRU ミドル API のコマンド X (A, B, C, D, E) に変換し、BRU ミドルウェア 180 に送り、BRU ユニット 50 を動作する。

【0117】

同様に、図 15 では、IO コントロール層 170 が、共通インタフェース (API) の CRW コマンド A (a, b) を、CRW ミドル API のコマンド A (a, b, c) に変換し、CRW ミドルウェア 180 に送り、CRW ユニット 10 を動作し、共通インタフェース (API) の BRU コマンド X (A, B) を、BRU ミドル API のコマンド X (A, B, C, D, E, F) に変換し、BRU ミドルウェア 180 に送り、BRU ユニット 50 を動作する。

【0118】

図 16 の通帳処理コマンドを例に、具体例で説明する。ユニット初期化コマンドに対しては、ベンダ固有インタフェース (パラメータ) として、通帳プリンタのユーザ種別コード、通帳リボンニアエンドチェック有無の指定、通帳 MS (磁気ストライプ) エラーの休止有無の指定、通帳ページマーク上下使用の有無の指定、取り忘れ通帳引き込み時のページ閉じの有無の指定があり、各ベンダの通帳ユニットの仕様により、指定できる。

【0119】

通帳挿入処理コマンドに対しては、共通インタフェース (パラメータ) として、挿入媒体論理種別の指定、通帳タイプ番号 No. の指定、W-MS (ワイド磁気ストライプ) 切り替え指定、ストライプ位置指定、MS 消去の有無指定、指定行位置指定、複合動作指定、カメラ制御 (通帳挿入時にカメラを動作する) の指定がある。

【0120】

一方、ベンダ固有インタフェース (パラメータ) として、MS 位置の指定 (例えば、金融機関毎に通帳の MS の位置が異なる場合がある)、MS 方式 (記録フォーマット)、MS パラメータ (MS のデータ消去可か等)、ラインランプ制御の指定 (通帳入口にラインランプがある時に、挿入、排出時にランプを点灯する) がある。

【0121】

図 17 は、前述のユニット初期化処理のフロー図である。

【0122】

(S10) ユニット初期化コマンドを受けると、パラメータファイル 160 より、初期化に関するベンダ固有の設定情報 (パラメータ) を読み込む。

【0123】

(S12) 読み込んだパラメータを初期化コマンドに付して、ユニットに送信し、応答を受信する。

【0124】

図 18 は、前述の通帳挿入処理のフロー図である。

【0125】

(S20) 通帳挿入コマンドを受けると、パラメータファイル 160 より、通帳挿入に

関するベンダ固有の設定情報（パラメータ）を読み込む。

【0126】

(S22) 読み込んだパラメータと、共通インタフェースから受信したパラメータ（入力情報）とから、ベンダ固有ユニットのパラメータを編集する。

【0127】

(S24) 編集したパラメータを通帳挿入コマンドに付して、ユニットに送信し、応答を受信する。

【0128】

このようにして、予め、ベンダ固有のパラメータをパラメータファイル160に設定しておき、IOコントロール層170が、共通インタフェース（API）のコマンドのパラメータと、パラメータファイル160のベンダ固有のパラメータとを編集し、ATMミドルAPIのコマンドに変換し、ATMミドルウェアに送り、ベンダ固有のIOユニット10を動作するので、共通インタフェース（API）のコマンドで、ベンダ固有のIOユニットを動作できる。即ち、既存のATMミドルウェアを、標準インタフェースで動作するように、カスタマイズ化できる。

【0129】

【他の実施の形態】

次に、前述のAgent、メソッドの他の実施の形態を説明する。図19は、本発明の他の実施の形態のエージェントの説明図である。図19に示すように、この例では、同期Agentと、メソッド（プログラム）を合体し、同期Agentは、同期制御の初期化、メカリセット、紙幣／硬貨挿入、媒体同時排出、印字／繰り出し／MSライト／排出準備、入金返却、収納、強制排出／取込み、ユニット情報取得／取引状態設定／二面表示制御、入出金準備、強制補充、ジャムリセット、カード／通帳挿入を有する。

【0130】

合体された各エージェント/メソッドは、図19の黒丸で示すIOコントローラ（即ち、IOユニット）にコマンドを発行し、コマンド実行結果の応答を受ける。例えば、同期制御の初期化エージェント/メソッドは、紙幣コントローラ175、硬貨コントローラ176、通帳コントローラ171、カードコントローラ172、レシートコントローラ174、ジャーナルコントローラ177、取引制御コントローラ178、テンキーコントローラ173（図5参照）に、初期化コマンドを発行し、各コントローラによる初期化処理を行わせ、各コントローラから初期化処理結果を応答として、受信する。

【0131】

同様に、同期制御のメカリセットエージェント/メソッドは、紙幣コントローラ175、硬貨コントローラ176、通帳コントローラ171、カードコントローラ172、レシートコントローラ174、ジャーナルコントローラ177（図5参照）に、メカリセットコマンドを発行し、各コントローラによるメカリセット処理を行わせ、各コントローラからメカリセット処理結果を応答として、受信する。

【0132】

以下、同様に、同期制御の紙幣／硬貨挿入エージェント/メソッドは、紙幣コントローラ175、硬貨コントローラ176（図5参照）に、挿入コマンドを発行し、各コントローラによる挿入処理を行わせ、各コントローラから挿入処理結果を応答として、受信する。

【0133】

図20は、本発明の他の実施の形態におけるWebサーバ100が、送信する画面コンテンツの説明図である。

【0134】

図20に示すように、HTML（ページ記述言語）で記述された画面コンテンツ内で、画面内で呼び出すAgent名を指定する。ここでは、一例として、アプレットタグ<APPLET CODE="U\_\_agtSync\_Initial.class">により、同期制御（Sync）の初期化エージェントを呼び出すことを示す。

## 【0135】

更に、<SCRIPT Language= "javaScript">の記述により、Java Scriptで、同期制御の初期化エージェントをメソッドとして呼び出しを指定する。即ち、ret=document. U\_agtSync\_\_initial. の記述で、同期制御の初期化エージェント/メソッドの呼び出しを指定する。尚、ここでは、画面表示コンテンツの詳細を省略してある。

## 【0136】

この画面コンテンツが、ブラウザ120にダウンロードされると、アプレットタグにより、同期制御の初期化エージェントが指定され、且つSCRIPTの呼び出しメソッド名により、同期制御の初期化エージェント/メソッドが呼び出される。

## 【0137】

前述のように、呼び出された初期化メソッドは、紙幣コントローラ175、硬貨コントローラ176、通帳コントローラ171、カードコントローラ172、レシートコントローラ174、ジャーナルコントローラ177、取引制御コントローラ178、テンキーコントローラ173（図5参照）に、初期化コマンドを発行し、各コントローラによる初期化処理を行わせ、各コントローラから初期化処理結果を応答として、受信する。

## 【0138】

この場合にも、図12に示したように、呼び出された初期化メソッドは、紙幣コントローラ175、硬貨コントローラ176、通帳コントローラ171、カードコントローラ172、レシートコントローラ174、ジャーナルコントローラ177、取引制御コントローラ178、テンキーコントローラ173（図5参照）に、初期化コマンドを連続的に発行し、これらコントローラからの応答を、処理完了したものから順次受ける。

## 【0139】

このため、I/Oユニットを並列的に制御でき、複数のI/Oユニットを制御しても、短時間で制御できる。これに比し、図23で示したように、従来のScriptで、I/Oユニット単位のメソッドを指定する方法では、Script毎に、コマンド発行、応答受信を順次行う順次制御のため、WebによるI/O制御に時間がかかる。

## 【0140】

更に、前述のAgentは、処理単位名が付された処理単位で設けられるため、この処理単位の内容を若干変更するだけで、各種の機能のATMに、同一のアプレットタグ、メソッド名で、対応できる。この例でも、前述のメソッド呼び出しSCRIPTのメソッド名の括弧内に設けられたinitial(input Param)の変更により実現している。即ち、図11に示したように、input Paramは、各I/Oコントローラ（紙幣、硬貨、通帳、カード、レシート、ジャーナル、取引制御、テンキー）に、入力フラグを設定する8ビットの情報で構成されている。

## 【0141】

入力フラグは、"1" がセットされたI/Oコントローラへのメソッドの入力を許可することを示す。呼び出されたメソッドは、この入力フラグを参照して、コマンドを発行するI/Oコントローラを決定する。このため、入力フラグをWebサーバ100で操作することにより、異なる機能のATMでも、同一のアプレットタグ、メソッド名で、同一の処理を実行させることができる。

## 【0142】

例えば、硬貨を扱わない取引装置では、硬貨の入力フラグを"0" にセットしておけば、硬貨コントローラへのコマンド発行を防止できる。同様に、通帳を扱わない取引装置では、通帳の入力フラグを"0" にセットしておけば、通帳コントローラへのコマンド発行を防止できる。

## 【0143】

この実施の形態は、画面コンテンツとともに送信するアプレット（エージェント）が、図7の例に比し、比較的小容量のプログラムに分割できるため、送信コンテンツのデータ量を少なくでき、通信時間を短縮できる。

## 【0144】

一方、図7のエージェントの例では、同期制御のアプレットが、多数のメソッドを搭載しているため、比較的大容量のプログラムであるが、ATM1で、アプレットをキャッシュすることにより、次回からこのアプレットの送信を省略でき、次回からの送信コンテンツのデータ量を少なくでき、通信時間を短縮できる。

【0145】

前述の実施の形態では、図1のような自動取引装置として、自動入出金機を例に、説明したが、出金機、自動現金貸出機、自動発行機等の他の装置に適用できる。又、ネットワークをインターネットで説明したが、他のネットワークに適用でき、且つサーバは、SCRIPTは、Java SCRIPTのみならず、他のSCRIPTのものにも適用できる。

【0146】

更に、共通APIによるカスタマイズ化したミドルウェア制御を例に説明したが、これを使用せず、ミドルウェア制御するものにも、適用できる。

【0147】

以上、本発明を実施の形態により説明したが、本発明の趣旨の範囲内において、本発明は、種々の変形が可能であり、本発明の範囲からこれらを排除するものではない。

【0148】

(付記1) 利用者の操作に応じて、Webサーバと交信し、案内表示と取引動作を行う自動取引装置において、前記案内表示を行う表示ユニットと、前記取引動作を行うための複数のIOユニットと、前記Webサーバからの画面コンテンツに応じて、前記表示装置の画面の案内表示を制御し、且つ前記画面コンテンツに埋め込まれたオブジェクトに応じて、前記複数のIOユニットを制御する制御ユニットとを有し、前記制御ユニットは、前記オブジェクトのスクリプトにより、前記複数のIOユニットを同期制御する前記取引動作の処理単位のメソッドを呼び出し、前記複数のIOユニットを同期制御することを特徴とする自動取引装置。

【0149】

(付記2) 前記制御ユニットは、前記Webサーバからの画面コンテンツを解釈し、前記案内表示を行うとともに、前記画面コンテンツに埋め込まれた前記オブジェクトのスクリプトを解釈し、前記複数のIOユニットを同期制御する前記取引動作の処理単位のメソッドを呼び出すブラウザを有し、前記ブラウザから前記複数のIOユニットを同期制御することを特徴とする付記1の自動取引装置。

【0150】

(付記3) 前記制御ユニットは、前記呼び出されたメソッドによるポスト依頼に応じて、前記Webサーバにリクエストを送信することを特徴とする付記1の自動取引装置。

【0151】

(付記4) 前記制御ユニットは、ページ記述言語で記述された画面作成プログラムと前記オブジェクトのスクリプトと、前記スクリプトで呼び出されるメソッドプログラムとを有する画面コンテンツを、前記Webサーバから受信することを特徴とする付記1の自動取引装置。

【0152】

(付記5) 前記制御ユニットは、前記呼び出したメソッドにより、前記複数のIOユニットに動作コマンドを発行し、前記IOユニットからの応答を受信することを特徴とする付記1の自動取引装置。

【0153】

(付記6) 前記複数のIOユニットは、少なくとも、現金処理ユニットと、カード処理ユニットとを有することを特徴とする付記1の自動取引装置。

【0154】

(付記7) 前記制御ユニットは、前記スクリプトに付記された入力パラメータに従い、前記メソッドにより同期制御する複数のIOユニットを指定することを特徴とする付記1の自動取引装置。

## 【0155】

(付記8) 前記ブラウザは、前記画面コンテンツのページ記述言語で記述された画面作成プログラムにより前記案内画面を作成するとともに、前記オブジェクトのスクリプトから、前記メソッドプログラムを呼び出し、前記複数のI/Oユニットを同期制御することを特徴とする付記4の自動取引装置。

## 【0156】

(付記9) 前記ブラウザは、前記画面コンテンツのページ記述言語で記述された画面作成プログラムにより前記案内画面を作成するとともに、前記オブジェクトのスクリプトのアプレット指定とメソッド指定から、前記アプレットのメソッドプログラムを呼び出し、前記複数のI/Oユニットを同期制御することを特徴とする付記8の自動取引装置。

## 【0157】

(付記10) 前記ブラウザは、前記呼び出したメソッドにより、前記I/Oユニットの各々を制御する複数のI/Oコントローラに、動作コマンドを発行し、前記I/Oコントローラからの応答を受信することを特徴とする付記5の自動取引装置。

## 【0158】

(付記11) Webサーバと、前記Webサーバにネットワークを介し接続され、利用者の操作に応じて、前記Webサーバと通信し、案内表示と取引動作を行う自動取引装置とを有し、前記自動取引装置は、前記案内表示を行う表示ユニットと、前記取引動作を行うための複数のI/Oユニットと、前記Webサーバからの画面コンテンツに応じて、前記表示装置の画面の案内表示を制御し、且つ前記画面コンテンツに埋め込まれたオブジェクトに応じて、前記複数のI/Oユニットを制御する制御ユニットとを有し、前記制御ユニットは、前記オブジェクトのスクリプトにより、前記複数のI/Oユニットを同期制御する前記取引動作の処理単位のメソッドを呼び出し、前記複数のI/Oユニットを同期制御することを特徴とする自動取引システム。

## 【0159】

(付記12) 前記自動取引装置の前記制御ユニットは、前記Webサーバからの画面コンテンツを解釈し、前記案内表示を行うとともに、前記画面コンテンツに埋め込まれた前記オブジェクトのスクリプトを解釈し、前記複数のI/Oユニットを同期制御する前記取引動作の処理単位のメソッドを呼び出すブラウザを有し、前記ブラウザから前記複数のI/Oユニットを同期制御することを特徴とする付記11の自動取引システム。

## 【0160】

(付記13) 前記制御ユニットは、前記呼び出されたメソッドによるポスト依頼に応じて、前記Webサーバにリクエストを送信することを特徴とする付記11の自動取引システム。

## 【0161】

(付記14) 前記Webサーバは、ページ記述言語で記述された画面作成プログラムと前記オブジェクトのスクリプトと、前記スクリプトで呼び出されるメソッドプログラムとを有する画面コンテンツを、前記自動取引装置に送信することを特徴とする付記11の自動取引システム。

## 【0162】

(付記15) 前記制御ユニットは、前記呼び出したメソッドにより、前記複数のI/Oユニットに動作コマンドを発行し、前記I/Oユニットからの応答を受信することを特徴とする付記11の自動取引システム。

## 【0163】

(付記16) 前記複数のI/Oユニットは、少なくとも、現金処理ユニットと、カード処理ユニットとを有することを特徴とする付記11の自動取引システム。

## 【0164】

(付記17) 前記Webサーバは、前記スクリプトに、前記メソッドにより同期制御する複数のI/Oユニットを指定する入力パラメータを付記して、前記自動取引装置に送信し、前記制御ユニットには、前記スクリプトに付記された入力パラメータに従い、前記メソ

ッドにより同期制御する複数の I O ユニットを指定することを特徴とする付記 11 の自動取引システム。

【0165】

(付記 18) 前記ブラウザは、前記画面コンテンツのページ記述言語で記述された画面作成プログラムにより前記案内画面を作成するとともに、前記オブジェクトのスクリプトから、前記メソッドプログラムを呼び出し、前記複数の I O ユニットを同期制御することを特徴とする付記 14 の自動取引システム。

【0166】

(付記 19) 前記ブラウザは、前記画面コンテンツのページ記述言語で記述された画面作成プログラムにより前記案内画面を作成するとともに、前記オブジェクトのスクリプトのアプレット指定とメソッド指定から、前記アプレットのメソッドプログラムを呼び出し、前記複数の I O ユニットを同期制御することを特徴とする付記 18 の自動取引システム。

【0167】

(付記 20) 前記ブラウザは、前記呼び出したメソッドにより、前記 I O ユニットの各々を制御する複数の I O コントローラに、動作コマンドを発行し、前記 I O コントローラからの応答を受信することを特徴とする付記 15 の自動取引システム。

【産業上の利用可能性】

【0168】

このように、Web サーバの画面コンテンツに取引処理の各処理単位の Agent を埋め込み、処理単位の 1 つのメソッドを呼び出すことにより、複数の I O ユニットを動作制御でき、一般的な取引処理の流れで、異なる自動取引装置に共通に使用できる。このため、異なる機能、構成の自動取引装置を Web 制御する画面コンテンツの作成が容易となり、且つ並列制御が可能となり、高速の I O ユニット制御を実現できる。これにより、利用者の自動取引装置の待ち時間を短縮でき、且つ稼働率を向上でき、Web 制御の自動取引装置の普及に寄与する。

【図面の簡単な説明】

【0169】

【図 1】 本発明の一実施の形態の自動取引装置の外観図である。

【図 2】 図 1 の自動取引装置のブロック図である。

【図 3】 本発明の一実施の形態の自動取引システムのシステム構成図である。

【図 4】 図 3 のブラウザ及び画面コンテンツの説明図である。

【図 5】 図 3 及び図 4 の ATM ミドルウェアのブロック図である。

【図 6】 図 3 及び図 5 の共通インタフェースの取引コマンドの説明図である。

【図 7】 図 3 及び図 4 のエージェントの説明図である。

【図 8】 図 3 及び図 4 の他のエージェントの説明図である。

【図 9】 図 3 及び図 4 の画面コンテンツの説明図である。

【図 10】 図 3 及び図 4 のエージェントによる I O 制御動作の説明図である。

【図 11】 図 9 の画面コンテンツの入力パラメータの説明図である。

【図 12】 図 10 のエージェントのメソッドの動作説明図である。

【図 13】 本発明の一実施の形態の入金処理の説明図である。

【図 14】 本発明の一実施の形態のミドルウェア API 変換動作説明図である。

【図 15】 本発明の他の実施の形態のミドルウェア API 変換動作説明図である。

【図 16】 本発明の一実施の形態の共通インタフェースとベンダ固有インタフェースの説明図である。

【図 17】 I O コントロールライブラリでの初期化インタフェース変換処理フロー図である。

【図 18】 I O コントロールライブラリでの通帳挿入処理のインタフェース変換処理フロー図である。

【図 19】 本発明の他の実施の形態のエージェントの説明図である。

【図 20】本発明の他の実施の形態の画面コンテンツの説明図である。

【図 21】従来の Web 制御による自動取引システムの説明図である。

【図 22】従来の Web 制御の画面コンテンツの説明図である。

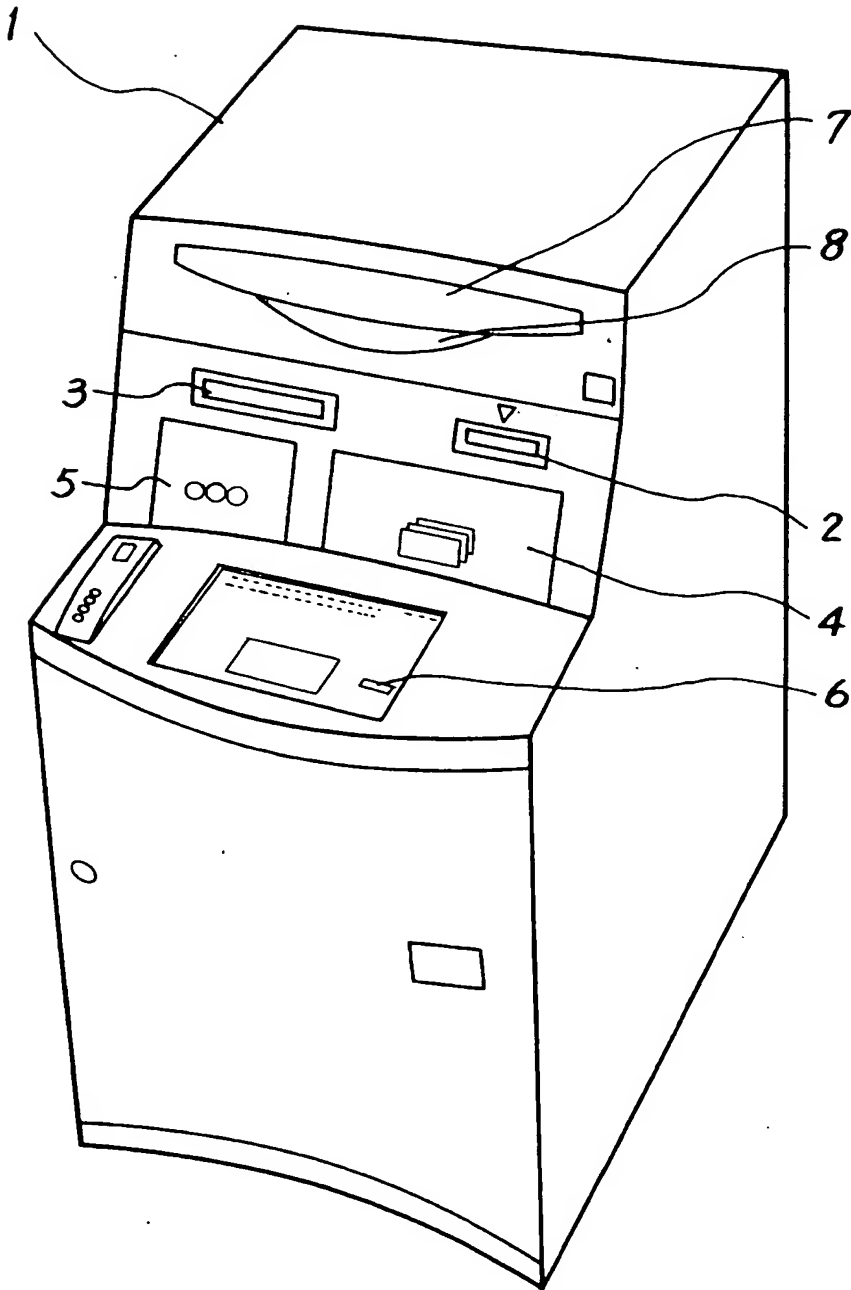
【図 23】従来の画面コンテンツのスク립トによる IO 制御動作のシーケンス図である。

【符号の説明】

【0170】

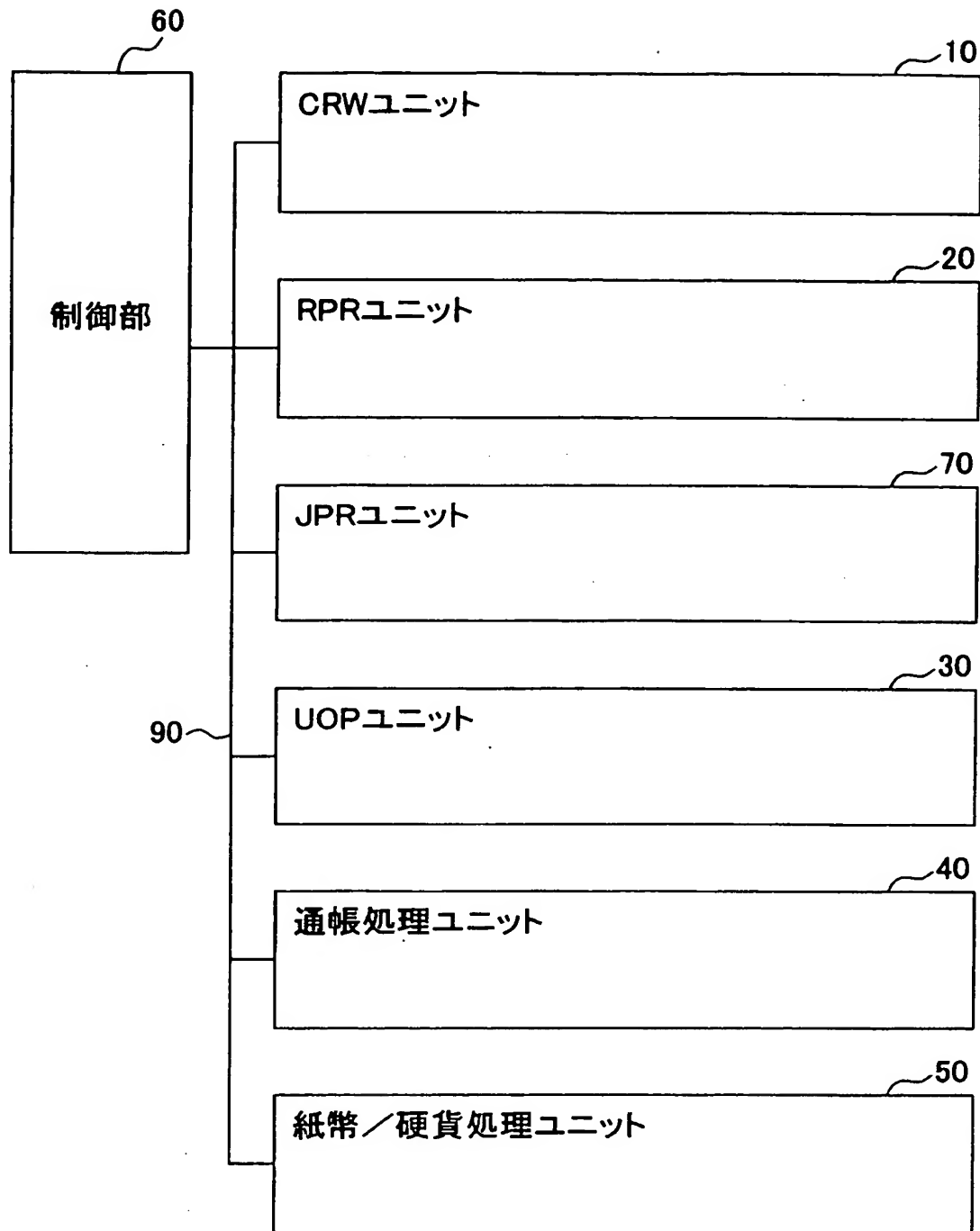
- 1 自動取引装置
- 2 カード出入口
- 3 通帳出入口
- 4 紙幣出入口
- 5 硬貨出入口
- 6 UOP
- 10 磁気カードリーダユニット
- 20 レシートプリンタユニット
- 30 UOP ユニット
- 40 通帳処理ユニット
- 50 紙幣／硬貨処理ユニット
- 60 制御ユニット
- 70 ジャーナルプリンタユニット
- 100 Web サーバ
- 110 ネットワーク
- 120 ブラウザ
- 122 画面コンテンツ
- 124、126、128 エージェント
- 130 ATM ミドルウェア
- 140 カーネル
- 150 デバイスドライバ
- 160 パラメータファイル
- 170 IO コントロール層
- 180 既存ミドルウェア層

【書類名】図面  
【図1】

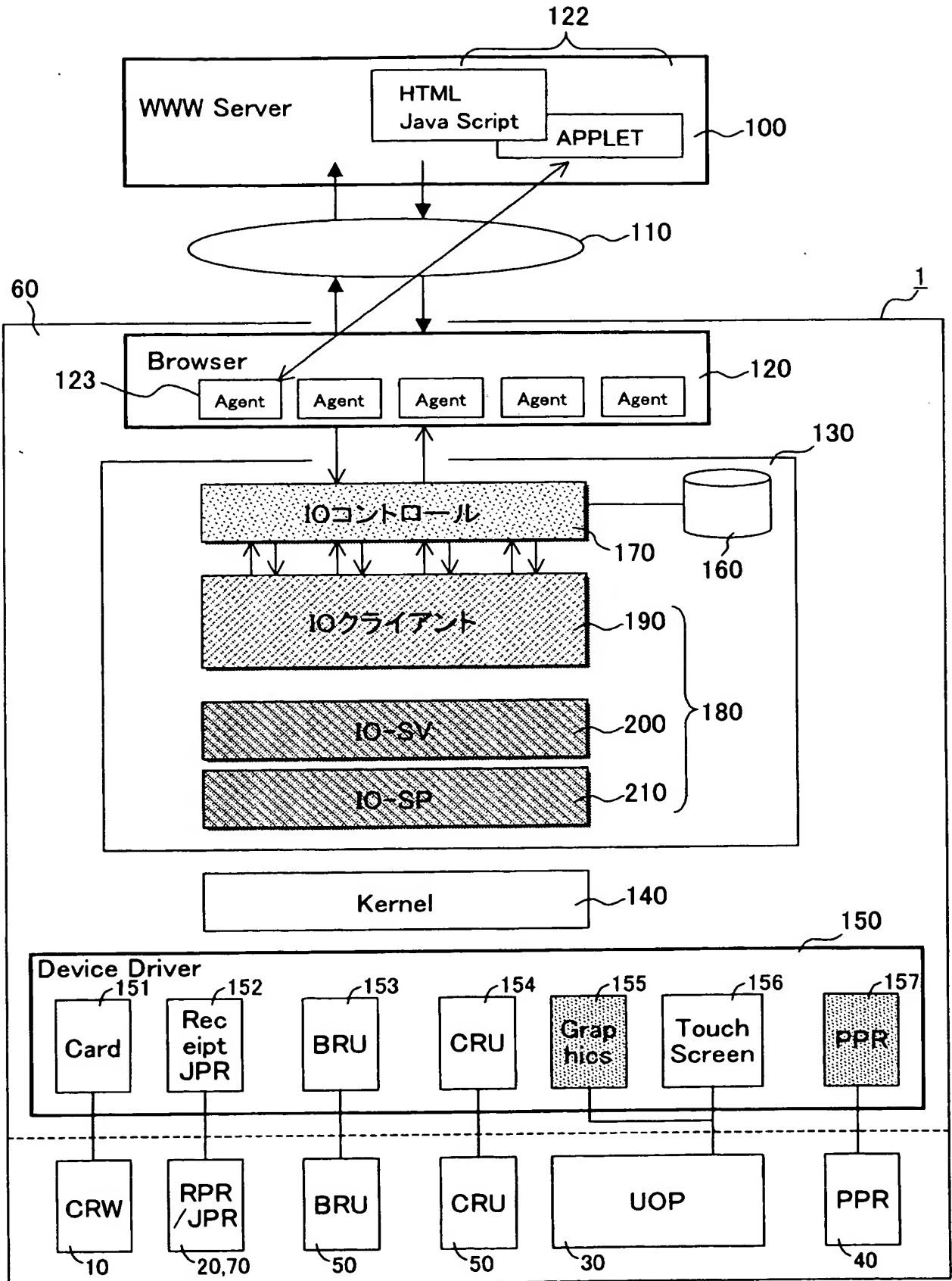




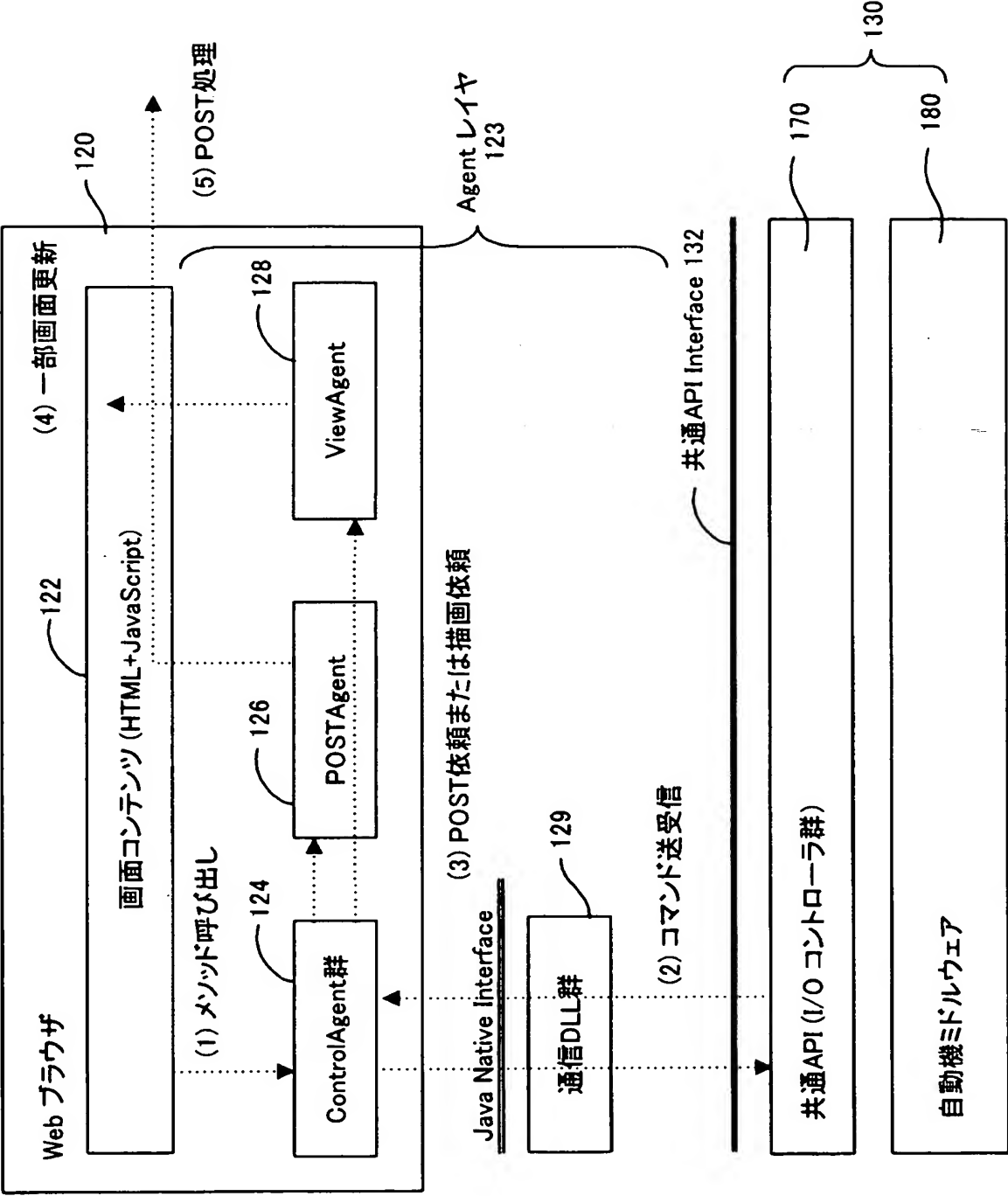
【図 2】



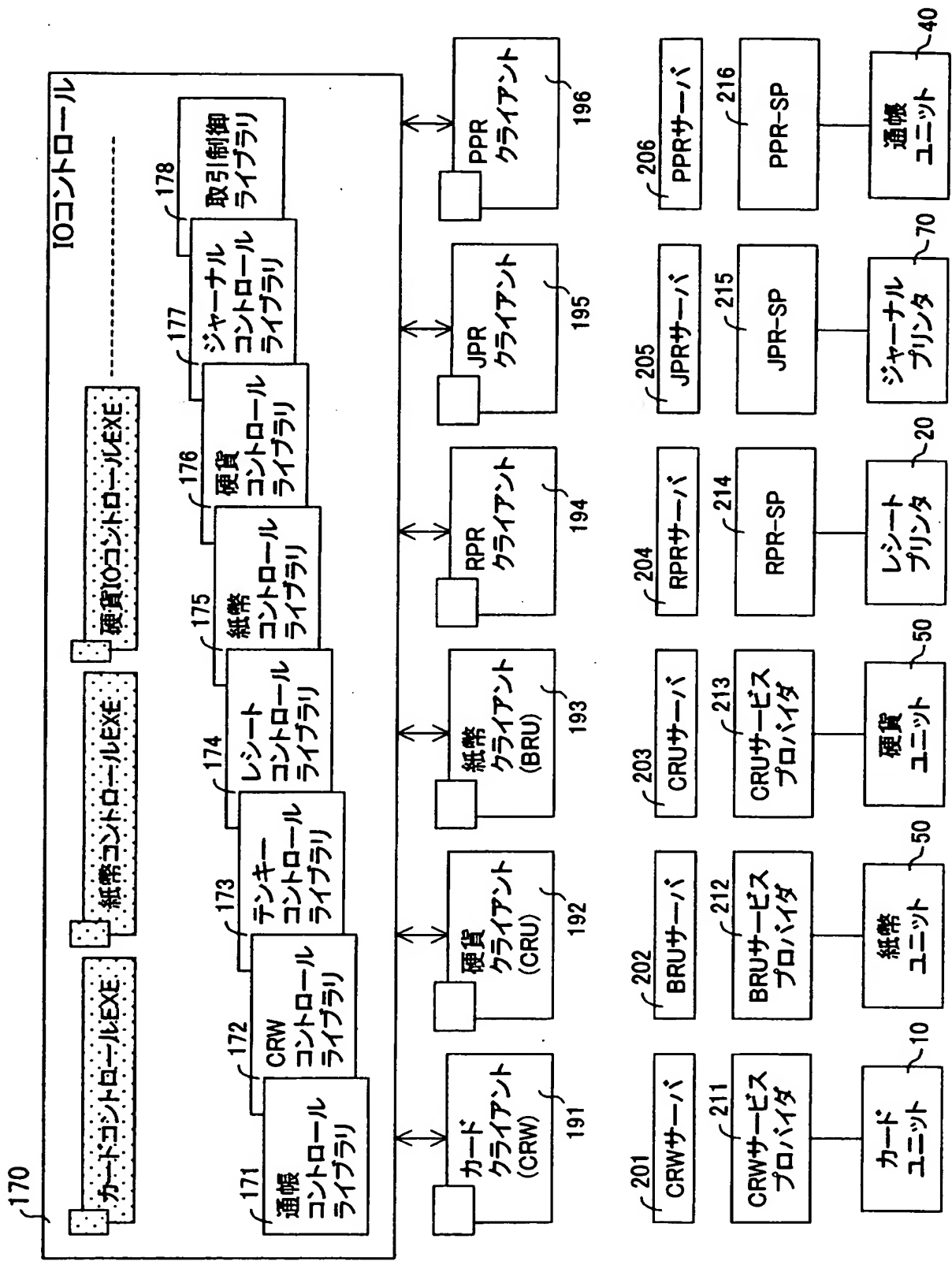
【図 3】



【図 4】



【図 5】



【図 6】

コマンド種別	コマンド
CRW コマンド	カード挿入
	カード排出
RPR コマンド	印字
	放出
PPR コマンド	通帳挿入
	印字
	MS書き込み
	通帳排出
	オートターンページ
BRU コマンド (CRU コマンド)	初期化
	受付／計数
	収納
	入金返却
	繰出
	放出
	取込
	搬送路チェック
	ジャムリセット

【図 7】

Agent	method	対象IOコントローラ								
		なし	紙幣	硬貨	通帳	カード	レシート	ジャーナル	取引	テンキー
同期制御	初期化		●	●	●	●	●	●	●	●
	メカリセット		●	●	●	●	●	●		
	紙幣／硬貨挿入		●	●						
	媒体同時排出		●	●						
	印字／繰出／MSライト／排出準備		●	●	●	●	●			
	入金返却		●	●						
	収納		●	●						
	強制排出／取込み		●	●	●	●	●		●	●
	ユニット情報取得/取引状態設定/二面表示制御		●	●	●	●	●	●	●	●
	入出金準備		●	●						
POST	強制補充		●	●						
	ジャムリセット		●	●						
	カード／通帳挿入				●	●				
	POST処理	●								
テキスト表示	POSTデータ保持	●								
	フォント設定	●								
	テキスト表示	●								
	テキスト消去	●								

【図 8】

Agent	method	対象IOコントロールラ						
		なし	紙幣	硬貨	通帳	カード	レシート	ジャーナル
紙幣制御	受付／計数		●					
	収納		●					
	入金返却		●					
	放出		●					
	キャンセル		●					
通帳制御	行セット／ペーシマ－グリード				●			
	MSリード				●			
	オートターンペーシ				●			
	ページチェックオートターン				●			
	通帳構成情報設定				●			
カード制御	カード挿入					●		
	キャンセル					●		
	振込みカード印字					●		
	振込みカード発行					●		
	排出準備					●		
レシート制御	オーバ－レイ登録						●	
	排出準備						●	
取引制御	取引情報設定							●
	装置状態監視							●
	装置状態取得							●
	運用情報設定							●
	キャンセル							●
テンキー	テンキー入力開始							●
	テンキー入力終了							●

【図 9】

```

<HTML>
<HEAD>
<TITLE>WebATM Sample</TITLE>
</HEAD>

<BODY>

// 画面内で呼び出す Agent を指定する.
<APPLET CODE="U_agtSync.class" codebase = "http://webatm:8080/webatm"
        WIDTH="0" HEIGHT="0" NAME="U_agtSync" >
</APPLET>

// POSTAgent が JavaScript の関数を呼び出せるように MAYSCRIPT 定義を行う.
<APPLET CODE="U_agtPost.class" codebase = "http://webatm:8080/webatm"
        WIDTH="0" HEIGHT="0" NAME="U_agtPost" MAYSCRIPT>

<SCRIPT language="javascript">
<!--
        // POSTAgent から呼び出される関数を定義する. (必須)
        function postFunc(postData){ document.sys_form.ioResponse.value = postData;
                                window.document.sys_form.submit();
        }

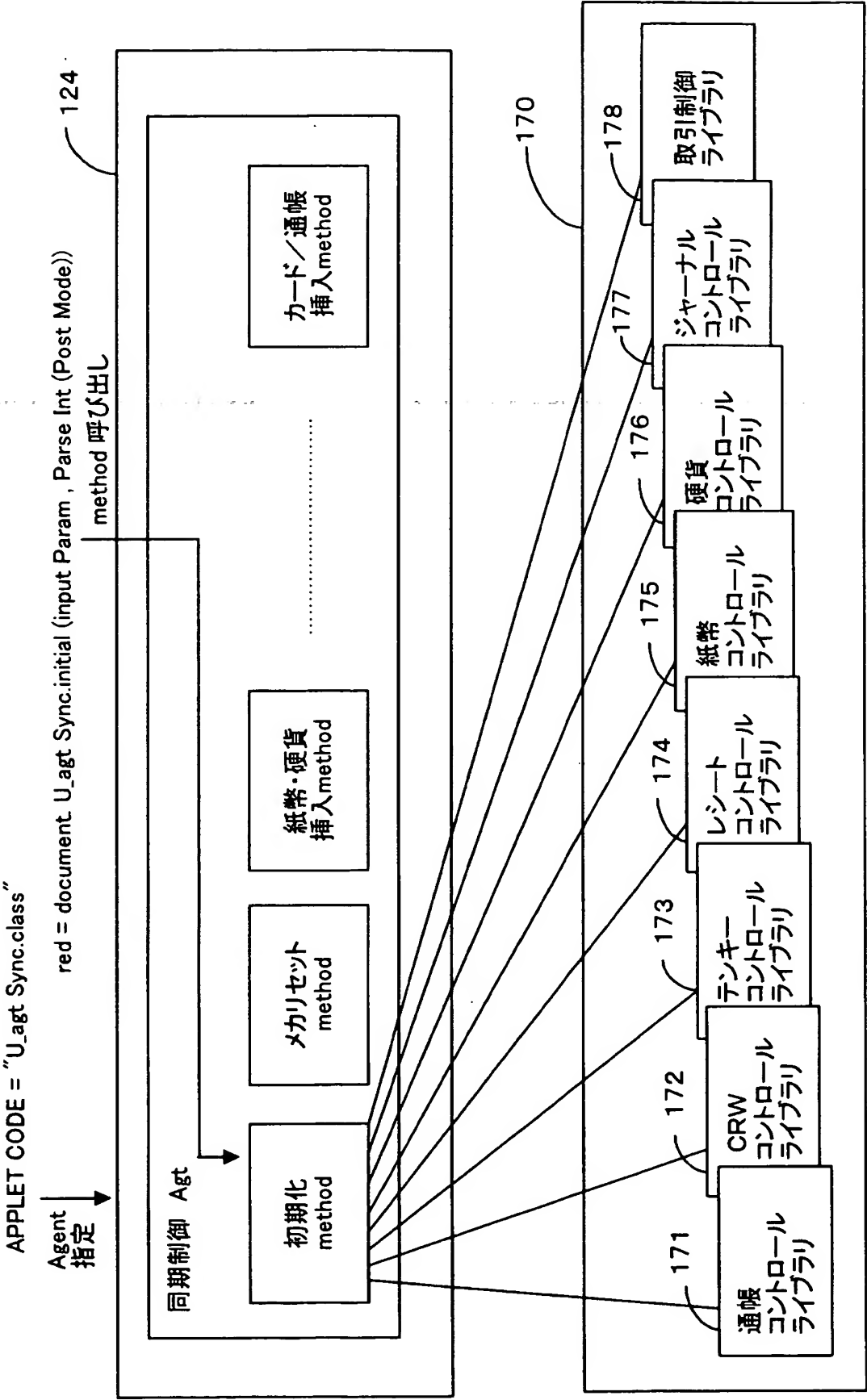
        // 同期 Agent の初期化メソッドの呼び出し
        ret = document.U_agtSync_initial (inputParam, parseInt(postMode), parseInt(maxTimer));
        if (ret < 0) {
                // 異常通知処理
                ~省略~
        }

// -->
</SCRIPT>
</BODY>
</HTML>

```



【図 10】



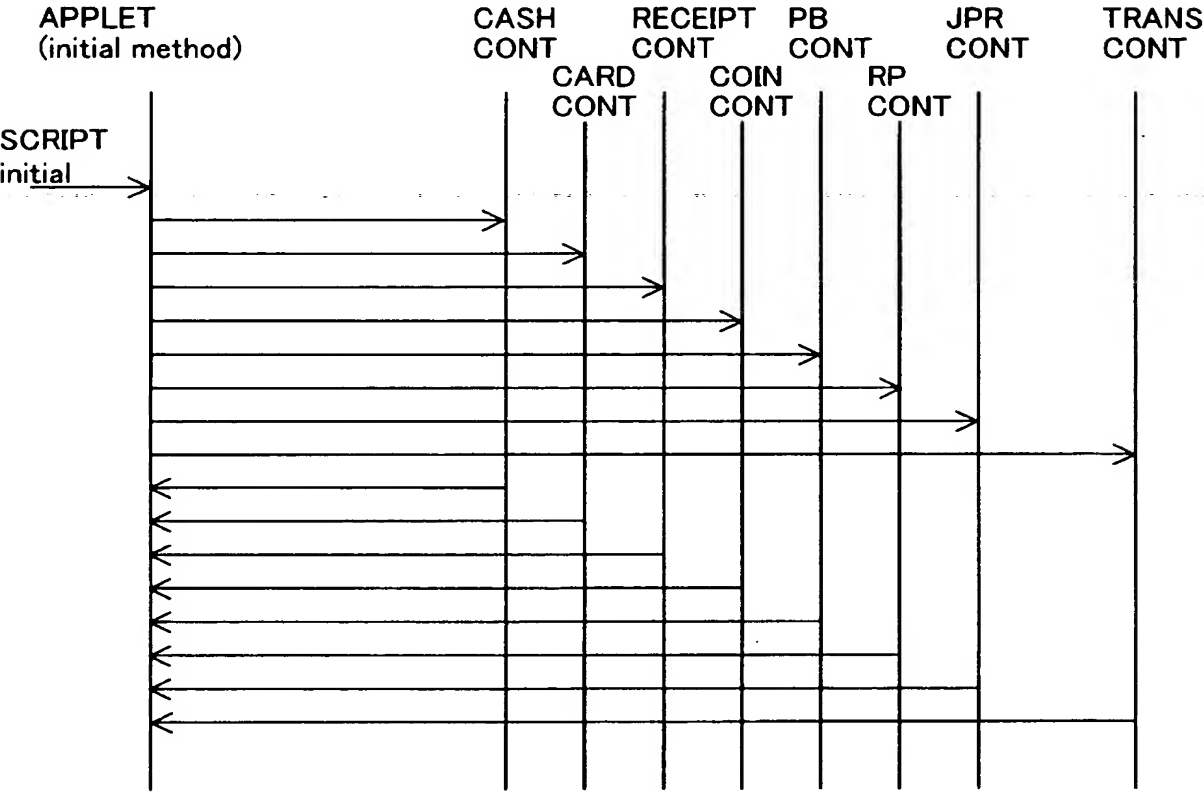
【図 1 1】

Input Param

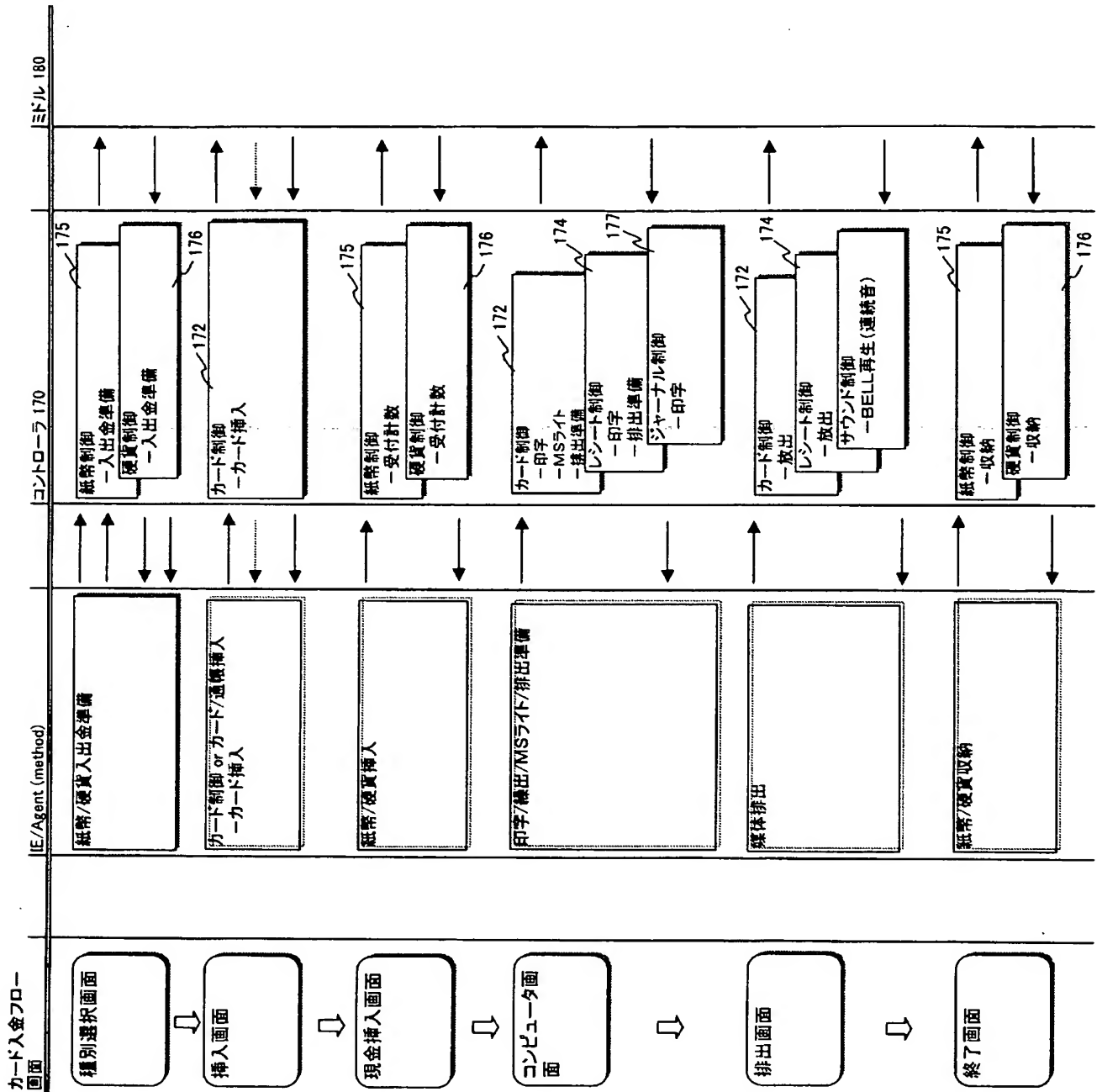
1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

紙幣 硬貨 通帳 カード レシートジャーナル 取引 テンキー

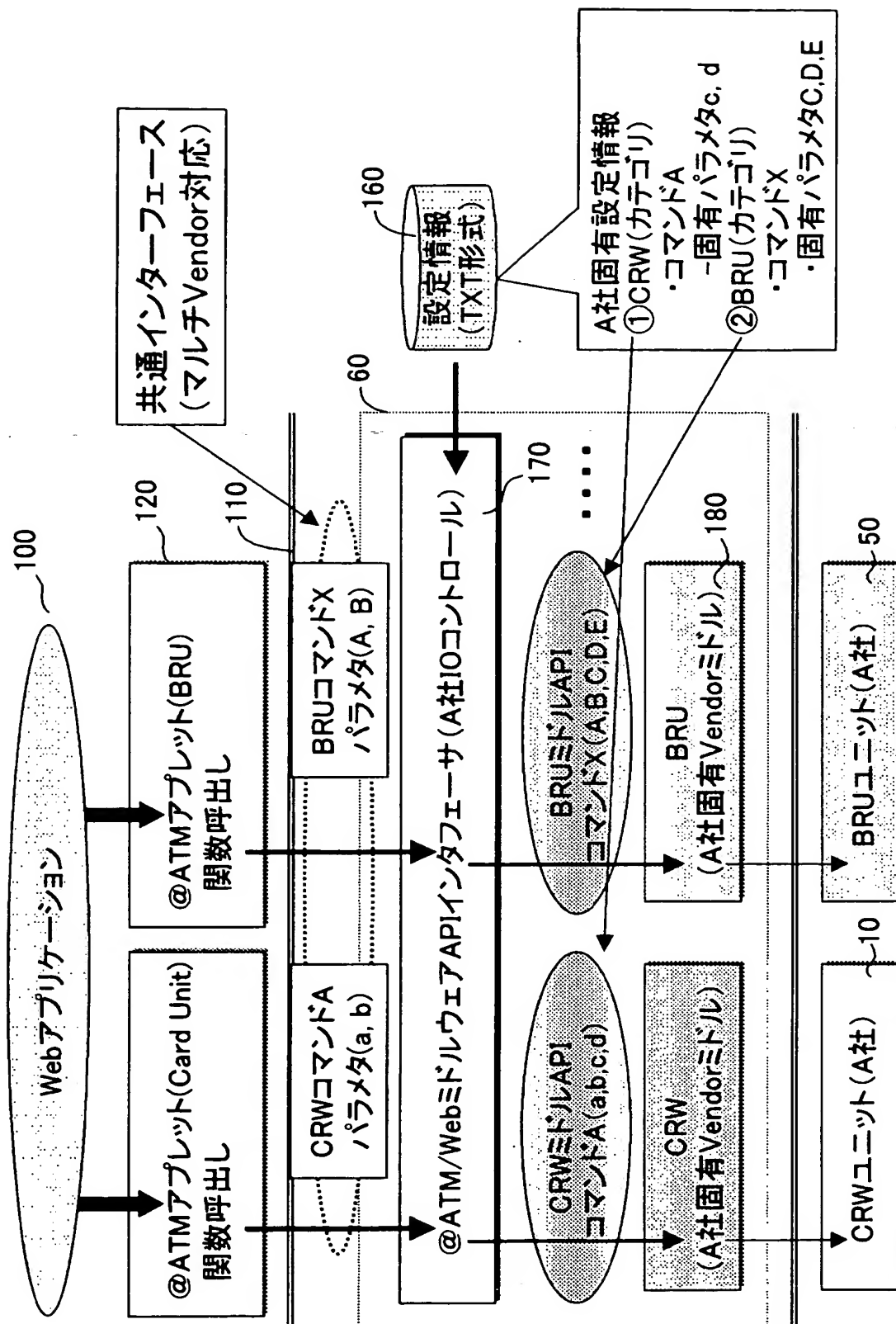
【図 1 2】



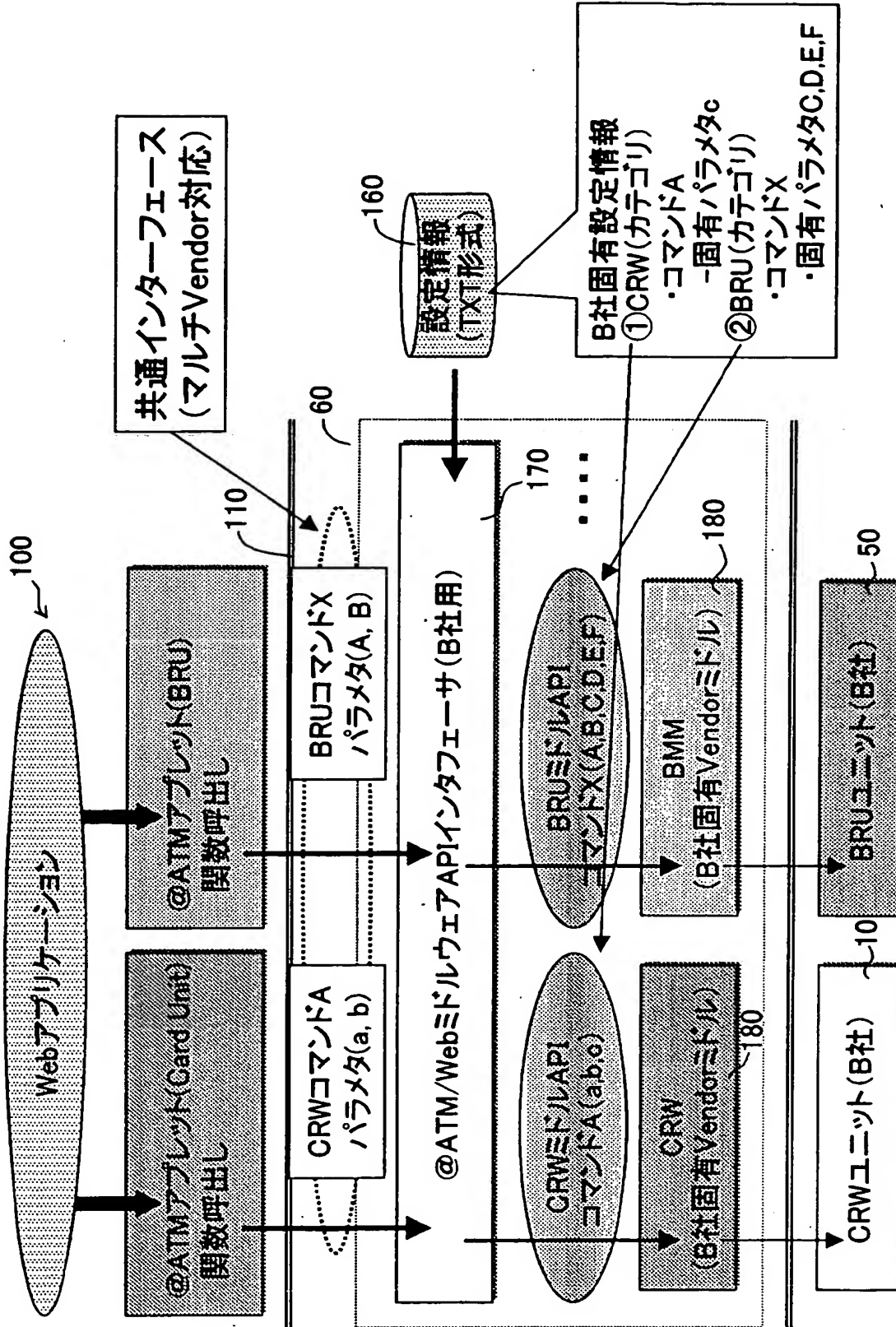
【図 13】



【図 14】



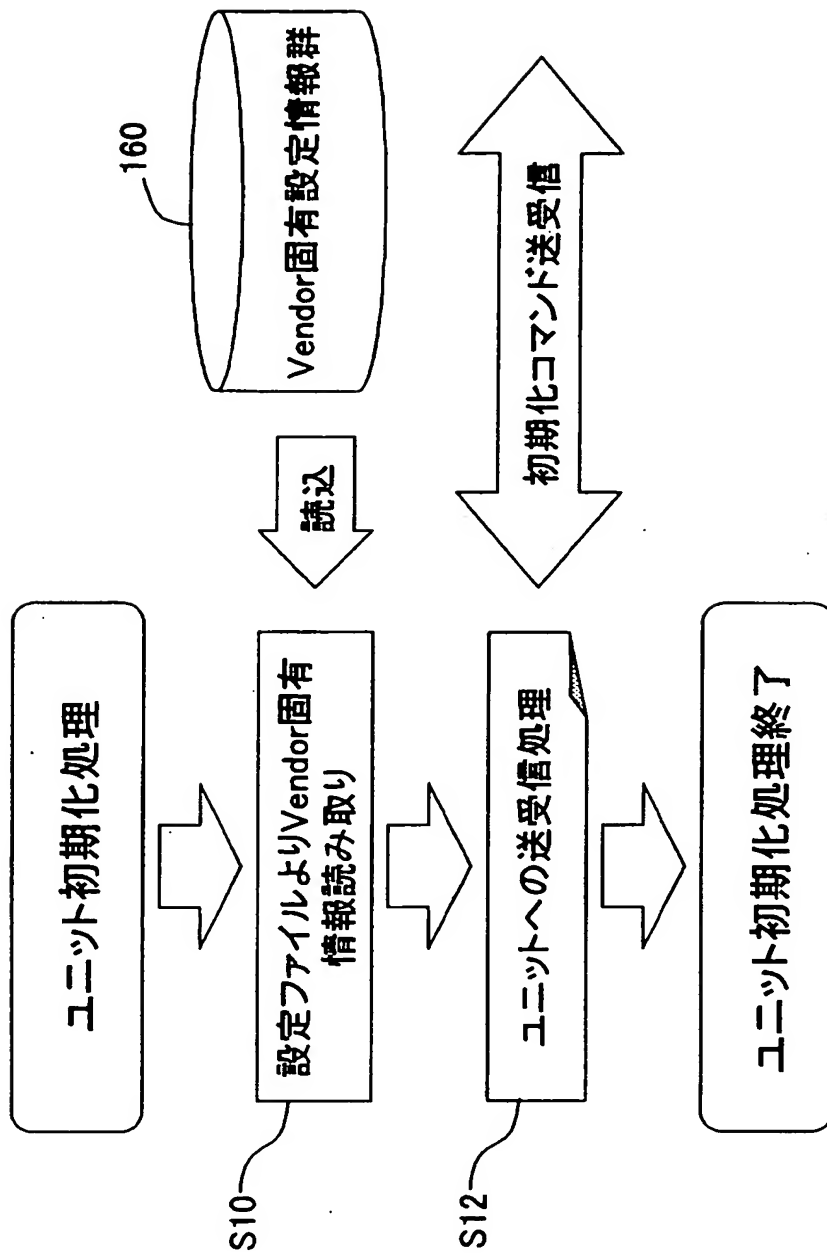
【図 15】



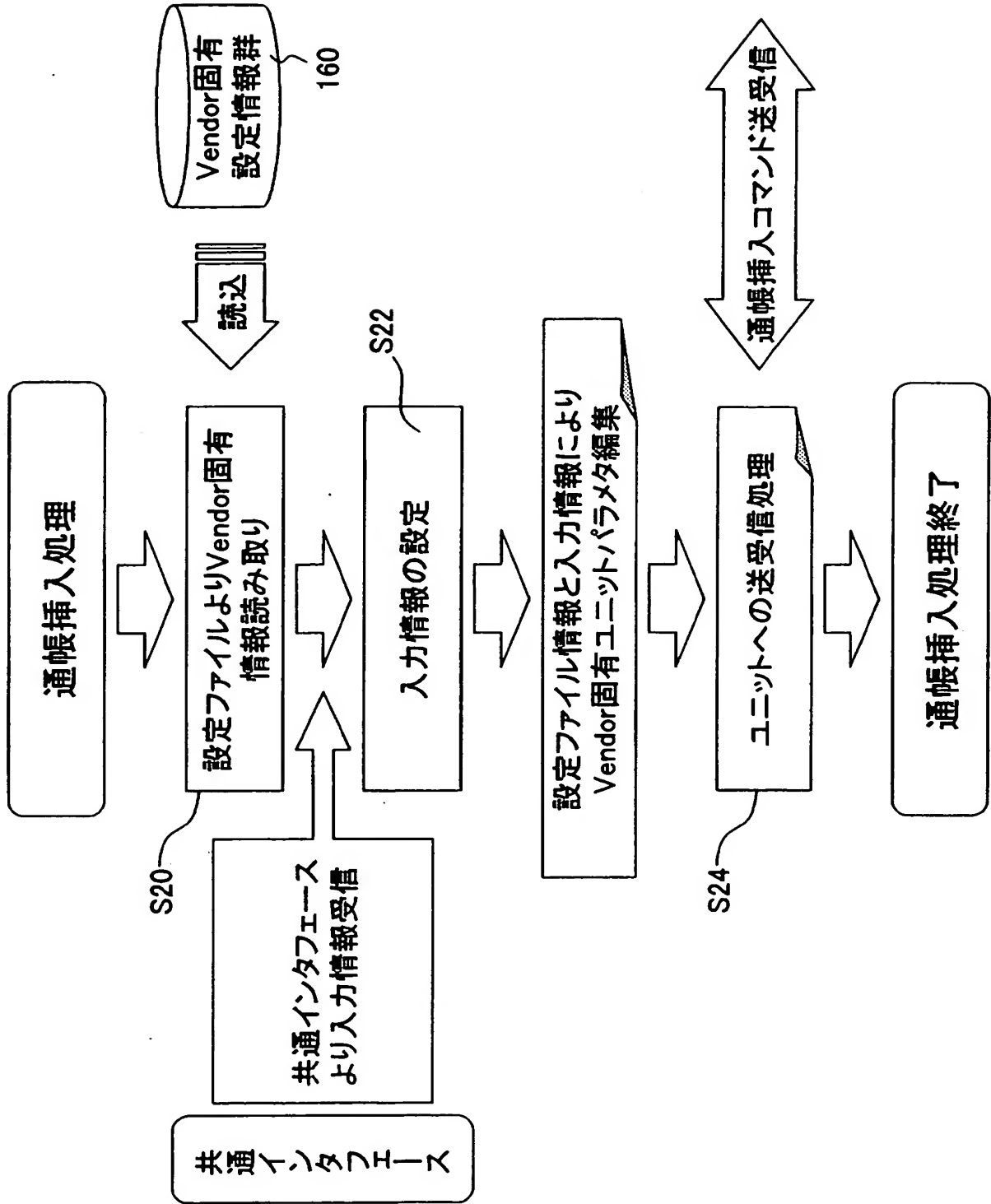
【図16】

項番	処理	共通インタフェース	ベンダ固有インタフェース
1	ユニット初期化処理		通帳プリンタユーザ種別コード
2			通帳リボンニアエントチェック指定
3			通帳MSライトエラー休止指定
4			通帳ページマーク上下使用指定
5			取り忘れ通帳引き込みページ閉じ指定
6	通帳挿入処理	挿入媒体論理種別指定	
7		通帳タイプNo.指定	
8		W-MS切り替え指定	
9		ストライプ位置指定	
10		MS消去指定	
11		指定行位置指定	
12		複合動作指定	
13		カメラ制御指定	
14			MS位置
15			MS方式
16			MSパラメタ
17			ラインランプ制御指定

【図 17】



【図18】





【図 19】

Agent (method)		対象IOコントロールラ									
		なし	紙幣	硬貨	通帳	カード	レシート	ジャーナル	取引	テンキー	
同期制御	初期化		●	●	●	●	●	●	●	●	
同期制御	メモリセット		●	●	●	●	●	●			
同期制御	紙幣／硬貨挿入		●	●							
同期制御	媒体同時排出		●	●	●		●				
同期制御	印字／繰出／MSライト／排出準備		●	●	●	●	●	●			
同期制御	入金返却		●	●							
同期制御	収納		●	●							
同期制御	強制排出／取込み		●	●	●	●	●				
同期制御	ユニット情報取得／取引状態設定／二面表示制御		●	●	●	●	●	●	●	●	
同期制御	キャンセル		●	●							
同期制御											
同期制御	入出金準備		●	●							
同期制御	強制補充		●	●							
同期制御	ジャムリセット		●	●							
同期制御	カード／通帳挿入				●	●					
POST	POST処理 POSTデータ保持	● ●									
テキスト表示	フォント設定 テキスト表示 テキスト消去	● ● ●									

【図 20】

```

<HTML>
<HEAD>
<TITLE>WebATM Sample</TITLE>
</HEAD>

<BODY>

// 画面内で呼び出す Agent を指定する.
<APPLET CODE="U_agtSync_initial.class" codebase = "http://webatm:8080/webatm"
        WIDTH="0" HEIGHT="0" NAME="U_agtSync_initial" >
</APPLET>

// POSTAgent が JavaScript の関数を呼び出せるように MAYSCRIPT 定義を行う.
<APPLET CODE="U_agtPost.class" codebase = "http://webatm:8080/webatm"
        WIDTH="0" HEIGHT="0" NAME="U_agtPost" MAYSCRIPT>

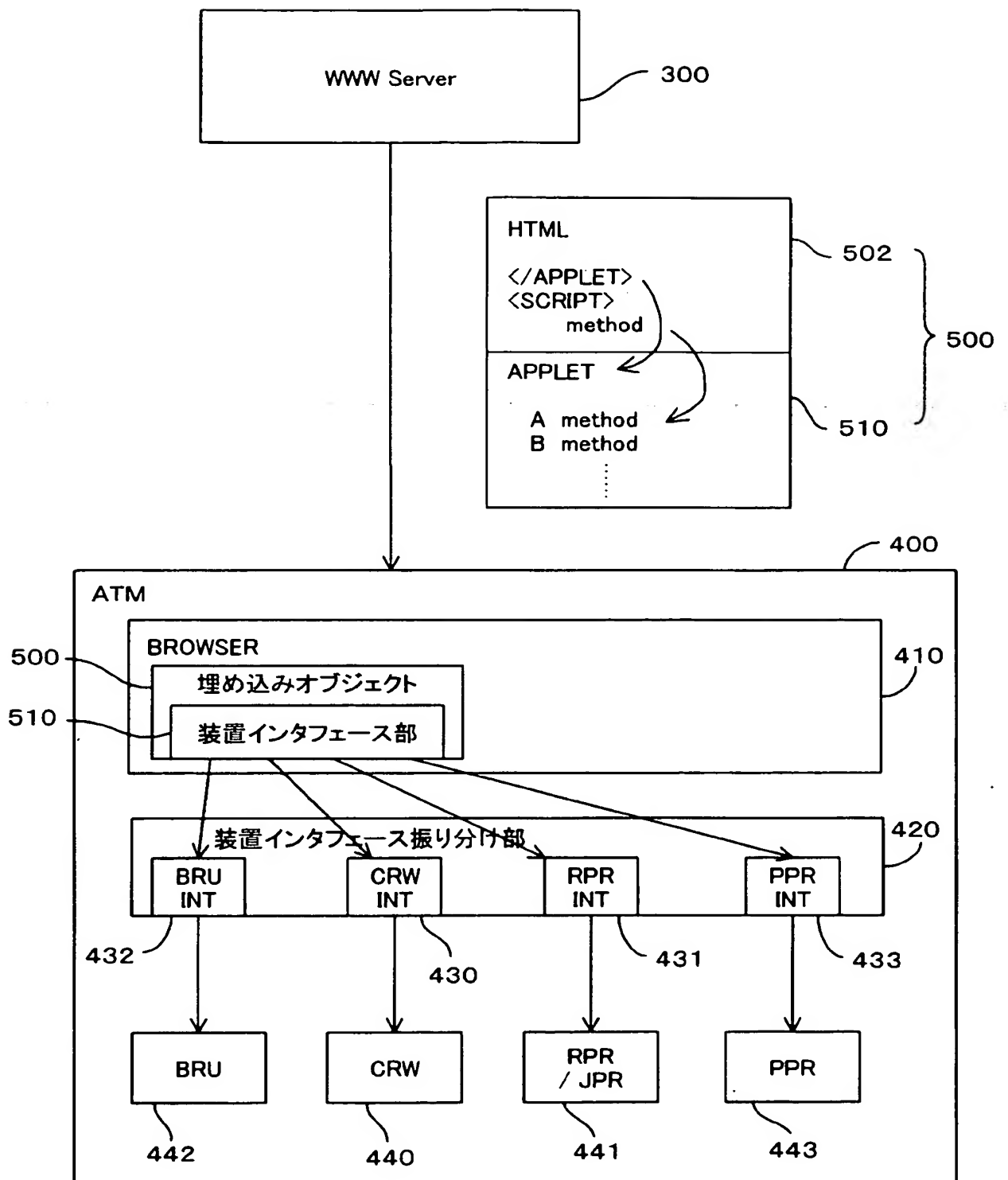
<SCRIPT language="javascript">
<!--
// POSTAgent から呼び出される関数を定義する. (必須)
function postFunc(postData){ document.sys_form.ioResponse.value = postData;
window.document.sys_form.submit();
}

// 同期 Agent の初期化メソッドの呼び出し
ret = document.U_agtSync_initial.initial (inputParam, parseInt(postMode), parseInt(maxTimer));
if (ret < 0) {
    // 異常通知処理
    ~省略~
}

// -->
</SCRIPT>
</BODY>
</HTML>

```

【図 21】



【図 2 2】

&lt;HTML&gt;

(表示画像のプログラム記述)省略

&lt;applet code=MachineIF.class codebase=MachineIF id=IDMachinesIF&gt;

&lt;/applet&gt; ..... (D 1)

&lt;script&gt;

// 各装置を初期化する。

FunctionInitialize{

IDMachineIF. Initialize("Cash");

IDMachineIF. Initialize("Card");

IDMachineIF. Initialize("Receipt");

IDMachineIF. Initialize("Passbook");

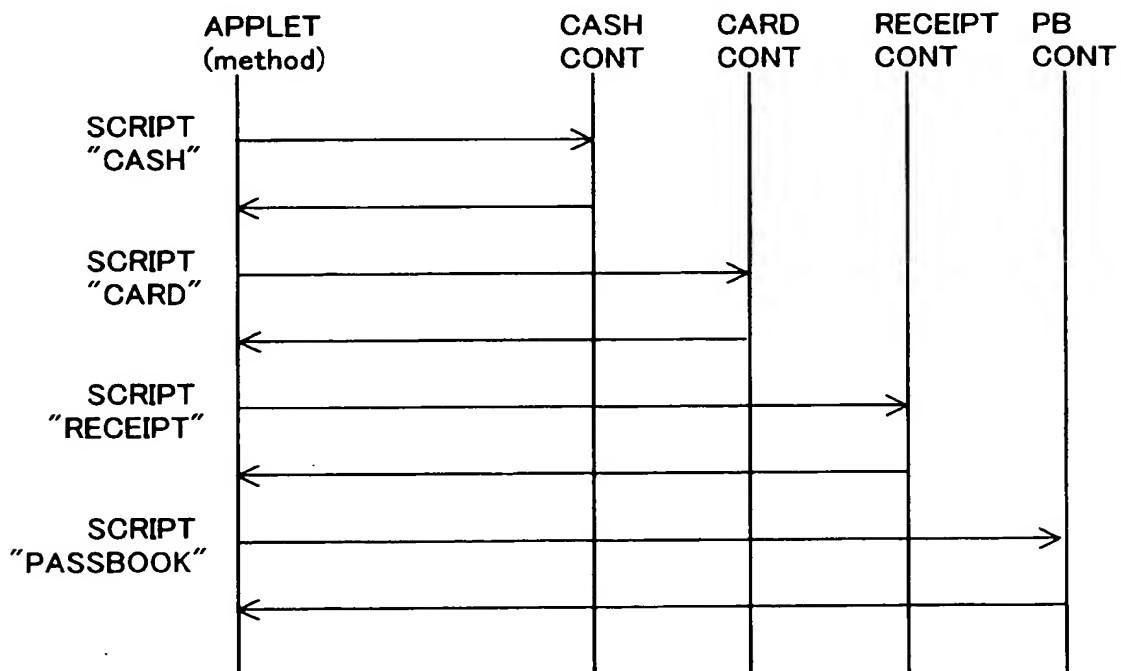
}

&lt;/script&gt;

(表示画像のプログラム記述)省略

&lt;/HTML&gt;

【図 2 3】



**【書類名】要約書****【要約】**

**【課題】** 利用者の操作に応じて、Webサーバと通信し、自動取引する自動取引装置に関し、異なる機能、ユニットを有する自動取引装置を制御する画面コンテンツの作成を容易とする。

**【解決手段】** Webサーバ(100)の画面コンテンツ(122)に取引処理の各処理単位のオブジェクト(124)を埋め込み、処理単位の1つのメソッドを呼び出して、複数のIOユニット(10～50)を動作制御する。一般的な取引処理の流れで、異なる自動取引装置に共通に使用でき、異なる機能、構成の自動取引装置をWeb制御する画面コンテンツの作成が容易となり、且つ並列制御が可能となり、高速のIOユニット制御を実現できる。

**【選択図】** 図4

特願 2 0 0 3 - 3 9 0 4 7 8

出 願 人 履 歴 情 報

識別番号 [ 0 0 0 0 0 5 2 2 3 ]

1. 変更年月日 1 9 9 6 年 3 月 2 6 日

[変更理由] 住所変更

住 所 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号

氏 名 富士通株式会社



特願 2 0 0 3 - 3 9 0 4 7 8

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 2 3 7 6 3 9 ]

1. 変更年月日

2 0 0 2 年 7 月 9 日

[変更理由]

名称変更

住 所

東京都稲城市矢野口 1 7 7 6 番地

氏 名

富士通フロンテック株式会社